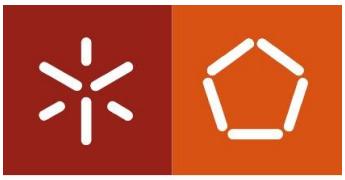

Universidade do Minho

Departamento de Informática

Trabalho Prático
Comunicação de Dados

Rui Filipe Pimenta Armada (a90468)

15 de fevereiro de 2021

Resumo

O presente relatório descreve o trabalho prático realizado no âmbito da disciplina de *Comunicação de Dados* (CD), ao longo do primeiro semestre, do segundo ano, do Mestrado Integrado em Engenharia Informática da Universidade do Minho.

O objetivo do projeto foi construir um sistema para comprimir ficheiros usando o algoritmo **LZW**, *Lempel–Ziv–Welch compression algorithm*.

Neste documento descrevemos sucintamente o sistema desenvolvido, as principais funções utilizadas, bem como as decisões tomadas durante o projeto.

Conteúdo

1 Introdução

O presente relatório foi elaborado no âmbito da unidade curricular *Comunicação de Dados* (CD), ao longo do primeiro semestre, do segundo ano, do Mestrado Integrado em Engenharia Informática da Universidade do Minho, e tem como objetivo a construção de um algoritmo capás de comprimir qualquer ficheiro aplicando o algoritmo *LZW* para diminuir o tamanho do ficheiro.

1.1 Descrição do Problema

A compressão *LZWd* é uma variante da compressão por padrões (ou por dicionário) conhecida como *LZW* (ver material de apoio). Dependendo do ficheiro, este tipo de algoritmo permite obter taxas de compressão e rendimentos superiores a 100, o que é impossível com algoritmos baseados em compressão estatística como o Shannon-Fano, Huffman ou codificação aritmética. Existem muitas variantes do algoritmo original *LZW* e que são usadas nos programas populares de compressão como o ZIP ou o RAR ou na construção dos formatos de ficheiros do tipo GIF ou PDF. A utilização deste tipo de algoritmo faz com que deixe de ser necessário usar o algoritmo RLE em ficheiros com muitas repetições consecutivas de símbolos pois estas sequências são casos especiais de padrões e são facilmente comprimidos pelos algoritmos baseados em *LZW*. O projeto tem como objetivo final Implementar um programa em C que, mediante um ficheiro de entrada, gere um ficheiro comprimido obtido por codificação *Lempel–Ziv–Welch-Dias(LZWd)*.

2 Implementação

Este trabalho consiste em realizar uma pré-compressão simples utilizando o mecanismo **LZW**, *Lempel–Ziv–Welch compression algorithm*. Este módulo receberá um ficheiro e irá guarda-lo por blocos dentro da memória. Um bloco poderá ter um tamanho até 64KBytes.

2.1 Função Principal

Este módulo gira em torno da função principal criada para gerir o trabalho é a função "call-lzwd". Esta função recebe o ficheiro no qual será aplicada a compressãoe vai aplicar o algoritmo LZW seguidamente de escrever o output que é gerado após a compressão num ficheiro ".txt.lzwd".

2.2 Compressão do ficheiro

A função que trata da compressão do ficheiro é a função "algorithm"que recebe como argumentos: o ficheiro, um array de duas dimensões que será o nosso Dicionario e o tamanho do Bloco. Aqui o programa irá comprimir o ficheiro utilizando a compressão **LZW**, *Lempel–Ziv–Welch compression algorithm*. No final da compressão esta função irá chamar a função responsavel por escrever o que está armazenado no dicionario, que será o que foi gerado pela compressão, e irá escrever para um ficheiro ".txt.lzwd".

2.3 Escrita da compressão LZWD

A função "translate"é a função responsavel pela escrita do output gerado pela função "algorithm"e que se encontra armazenado no dicionário. Esta função irá escrever para um ficheiro de output ".txt.lzwd"todo o output armazenado no dicionário, de acordo com o que foi proposto no enunciado.

2.4 Finalização

Posteriormente a isto tudo, indicado anteriormente, iremos terminar o tempo de execução, imprimir todos os dados relativos ao projeto e libertar todos os espaços alocados à memória

3 Limitações

Devido à dificuldade que encontrei para gerir os tamanhos dos padrões consegui apenas gerir padrões com tamanho limitado, o que faz com que se o padrão ultrapassar esse limite de tamanho o programa crache.

4 Ficheiros Usados

Como explicado na introdução, os ficheiros que utilizamos para testar são, com a exceção dos source files, os ficheiros gerados pelos módulos anteriores. Utilizamos os seguintes ficheiros:

- 1 ficheiro txt (*teste.txt*)
- 1 ficheiro txt (*something.txt*)

5 Conclusão

Considero que consegui adaptar bem a este projeto e consegui atingir a maioria dos objetivos estabelecidos neste trabalho que me foi proposto. Tentei cumprir o que foi exigido conseguindo executar a maioria dos objetivos propostos pelo enunciado do projeto. Além disso,pelo que eu observei, nenhuma parte do programa possui *memory leaks* ou outras falhas de memória que possão por em causa o projeto. Quanto aos resultados obtidos pelo meu projeto, venho a concluir que estes são bastante satisfatórios. Porém continuo a achar que existe espaço para melhorar o código a nivel de funcionalidade e de organização.