Relatório Projeto 2

Bruno Aziz Spring Machado
Departamento de Informática
Universidade Federal do Paraná – UFPR
Curitiba, Brasil
brunoazizsm@gmail.com

***Resumo*—Projeto de codificação e decodificação de mensagens de acordo com um tipo de Cifra de Livro, chamada "Cifra de Beale".**

***Index Terms*—exemplo, relatório, estilo, boa escrita.**

I. Introdução

Neste projeto foi solicitado a criação de uma codificação, criação de um arquivo de chaves e dois tipos de decodificações. A primeira decodificação é feita somente com o arquivo de chaves, e a segunda decodificação é feita diretamente com um Livro. Usei para os testes o livro e a mensagem oferecidas na wiki do professor Maziero.

O meu projeto está separado em três arquivos principais, onde o 'main.c' se trata do principal dos três, os secundários são: 'listaBeale.c' e 'codifica_decodifica.c', os três serão explicados nas próximas seções.

E, por fim, o 'Makefile' que serve para a execução do programa.

A estrutura do meu programa é um vetor de listas (uma matriz, basicamente), onde o indice do vetor é o numero na tabela ASCII dos caracteres.

II. Adendos

No começo eu pensei em usar uma Hash, porém a sua implementação em C não seria tão simples quanto imaginei, só percebi isso quando comecei a fazer o trabalho de fato, com isso decidi fazer um vetor de listas, como foi recomendado em sala.

Eu tratei a ausência das letras no livro e no arquivo de chaves da seguinte forma, se caso a letra não existir eu trato ela como '-404' e imprimo "(?)"na decodificação, para o usuário codificar a mensagem dele com um livro que faça sentido.

III. Arquivo Main.c

Neste arquivo faço a inclusão dos arquivos de cabeçalho dos meus outros dois programas para uso de suas funções.

Neste arquivo principal eu trato os três modelos de execução solicitados.

- ./beale -e -b LivroCifra -m MensagemOriginal -o MensagemCodificada -c ArquivoDeChaves

-./beale -d -i MensagemCodificada -c ArquivoDeChaves -o MensagemDecodificada

-./beale -d -i MensagemCodificada -b LivroCifra -o MensagemDecodificada

Para a verificação de qual comando será utilizado, é usado o 'argc', com isso vemos a quantidade de argumentos passados, pode ser notado nas execuções a cima que o primeiro possui 10 e os outros dois possuem 8 argumentos. Para verificar essa diferença dos dois com 8, eu vejo se o quinto argumento é '-c' ou '-b'.

IV. Codificação e Criação do Arquivo de Chaves

Primeiramente o livro é aberto para leitura e é feita a verificação se ele foi aberto. Após isso, o vetor de listas com 128 posições é criado, são 128 posições para abranger todos os caracters necessários. A funcão que "popula"o vetor de listas é chamada e logo após ela é feita a criação do arquivo de chaves. Em seguida, o arquivo com a mensagem que deve ser codificada é aberto em modo leitura e a função que cria a mensagem codificada é chamada. E no fim do arquivo a limpeza das alocações e os arquivos são fechados.

V. Decodificação com Arquivo de Chaves

Primeiramente, a mensagem codificada que foi criada anteriormente é aberta em modo leitura. Após isso, o vetor de listas é criado com as mesmas 128 posições da tabela ASCII, e após isso é chamada a função que "popula"o vetor de listas com o arquivo de chaves. E a função que decodifica é chamada, essa função lê o vetor de listas e procura cada valor da mensagem codificada nas listas, assim a decodificação é feita. E, por fim, os frees são dados e os arquivos são fechados.

VI. Decodificação com o Livro

Esta decodificação eu consegui fazer de maneira mais simples e sem tantos problemas. No começo o livro é aberto, o vetor de listas populado e o arquivo com a mensagem codificada é aberta. A função que decodifica a mensagem é chamada e a função busca nas listas o que o numero codificado representa. No fim a limpeza é feita com o free e os arquivos fechados.

VII. Arquivo Lista Beale

Listing 1. Exemplo de código em C.

```
struct nodo_l {
    int elemento;
    int posicao;
    struct nodo_l* prox;
};
typedef struct nodo_l nodo_l_t;
```

Essa é nossa principal estrutura utilizada, a mesma usada em Programação 1 que representa a lista.