Ramo Estudantil IEEE - UEL

GUILHERME POSSARI (guilherme.possari@uel.br)
MURILLO DE MOURA FERRAZ (murillo.ferraz@uel.br)
LUCAS TATSUYA UEDA (lucas.tatsuya.ueda@uel.br)
GABRIEL PERES DE SOUZA (gabriel.peres.souza@uel.br)

RELATÓRIO FINAL:

CÓDIGO MORSE

Londrina
2023

Ramo Estudantil IEEE - UEL

GUILHERME POSSARI
MURILLO DE MOURA FERRAZ
LUCAS TATSUYA UEDA
GABRIEL PERES DE SOUZA

RELATÓRIO FINAL:

CÓDIGO MORSE

Relatório apresentado ao Ramo Estudantil
IEEE da Universidade Estadual de Londrina.

Diretor de Projetos: Nathan Andreani Netzel
Gestores de Projetos: Daniel Tresse Dourado, Levi Monteiro dos Santos

Londrina
2023

POSSARI, Guilherme. FERRAZ, Murillo de Moura. UEDA, Lucas Tasuya. SOUZA, Gabriel Peres de. **RELATÓRIO FINAL:** CÓDIGO MORSE. 2023. Número total de folhas: 13. Relatório apresentado ao Ramo Estudantil IEEE da Universidade Estadual de Londrina, Londrina, 2023.

RESUMO

O projeto do Código Morse consiste na utilização de linguagem C (arquivo, ponteiros, strings, matrizes, arrays, estruturas de repetição e condição) para a criação de um programa que possibilita a codificação do alfabeto latino para código morse e decodificação de código morse para alfabeto latino.

Palavras-chave: Morse. Codificação. Decodificação. Alfabeto Latino. Linguagem C.

OBJETIVOS

Realizar a programação de um software que possibilite a codificação e decodificação de Código Morse. Demonstrar o resultado com saída em um arquivo .txt.

Contato do Ramo: sb.uel@ieee.org
Institute of Electrical and Electronics Engineers – IEEE

SUMÁRIO

1 FERRAMENTAS UTILIZADAS

1.1 Linguagem C:

A linguagem C foi criada por Dennis Ritchie em 1972 para reescrever de forma portável o sistema operacional UNIX, que antes era escrito em assembly. Sua estrutura e seu nome provêm de uma linguagem anterior B, que era uma simplificação da linguagem de programação BCPL, escrita em 1966.

Algumas das principais características da linguagem C são:

Linguagem procedural, modular e estruturada, com tipagem estática de dados;

• geralmente é compilada para o código de máquina da plataforma alvo, gerando código compacto, eficiente e sem necessidade de um subsistema runtime de execução sofisticado;

• possui facilidades para acesso de baixo nível à memória, registradores e portas de E/S;

• extremamente portável, pode executar em plataformas de microcontroladores a supercomputadores;

• é muito utilizada para escrever software de sistema, como sistemas operacionais, compiladores, serviços de rede, interfaces gráficas, bancos de dados, editores de texto, jogos, etc;

• a maior parte das funcionalidades da linguagem provém de vastas bibliotecas, como a biblioteca padrão C.

Figura 1 - Linguagem C

```
int main(){

    char palavra[50];
    int tamanho, i, j;
    FILE *arquivo = fopen("morse.txt","w");
```

Fonte: O próprio autor.

1.2 Strings

Uma string é uma sequência de caracteres que permite representar nomes, endereços e outras informações textuais. Em C, strings são implementadas como vetores de caracteres terminados pelo caractere especial "\0". Este caractere deve ser considerado ao medir tamanho do vetor. As aspas duplas ("...") são usadas para declarar strings constantes.

Figura 2 - String

```
01. #include <stdio.h>
02.
03. int main()
04. {
05.     char string[50];
06.
07.     scanf("%[^\n]%*c", string);
08.
09.     return 0;
10. }
```

Fonte: <https://arquivos.respondeai.com.br/seo-mirror/theory/2023/4a72dfc7-eb7c-4b0a-aa95-8365d029392d.webp>

1.3 Ponteiros

Em um programa, cada variável tem um endereço, que indica sua localização na memória do computador, e um conteúdo, que é o valor armazenado naquela posição de memória. Geralmente os valores armazenados são escalares (inteiros, reais ou caracteres) ou não-escalares (vetores, matrizes e estruturas).

Variáveis do tipo ponteiro armazenam endereços de memória de outras variáveis; em outras palavras, ponteiros são variáveis que referenciam outras variáveis. Ponteiros podem ser muito úteis em determinadas situações, por isso são usados extensivamente na programação em C.

Figura 3 - Ponteiro

```
int   *pi ;  // ponteiro para inteiro
float *pf ;  // ponteiro para float
char  *pc ;  // ponteiro para caractere
```

Fonte: <https://www.inf.ufpr.br/roberto/ci067/08_pointers.html>

1.4 Estruturas de repetição (for)

A estrutura de repetição for na linguagem C é usada para criar loops, permitindo que um bloco de código seja executado repetidamente com base em uma condição e um controle de iteração. É especialmente útil quando se sabe de antemão quantas vezes o loop deve ser executado, pois permite a especificação clara da inicialização, condição e atualização em um único local. O loop continuará a ser executado enquanto a condição for verdadeira e terminará quando a condição se tornar falsa.

Figura 4 - For

```
for (inicialização; condição; atualização) {
    // Corpo do loop
    // Código a ser repetido
}
```

Fonte: O próprio autor.

1.5 Visual Studio Code

O Visual Studio Code é um editor de código-fonte gratuito e de código aberto desenvolvido pela Microsoft. Ele é amplamente utilizado por desenvolvedores de software para escrever, editar e depurar código em diversas linguagens de programação. O VS Code é conhecido por sua leveza, extensibilidade e suporte a uma ampla variedade de extensões, o que o torna uma ferramenta popular para desenvolvimento de software em muitos ambientes e plataformas.

Figura 5 - Visual Studio Code

Fonte: <https://miro.medium.com/v2/resize:fit:720/0*S0gllBsD11p4kfwO.png>

2 METODOLOGIA

2.1 Codificação

Inicialmente, para se armazenar o alfabeto latino e seu equivalente em código morse, foram criadas dois arrays de caracteres, "char letra" e "char morse", respectivamente. Após o usuário escolher a opção "codificar" no menu inicial, será exigida a inserção da palavra a ser codificada. O programa irá armazenar a palavra e verificar seu tamanho, assim, através de dois laços de repetição,a percorre e analisa individualmente cada letra, já transformando-a imediatamente em seu equivalente em morse.

O processo é dado como encerrado quando o arquivo .txt com a codificação é criado e fica acessível ao usuário.

Figura 6 - Arrays

```
char letra[37] = {'a', 'b', 'c', 'd',
                  'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm',
                  'n', 'o', 'p', 'q', 'r', 's', 't',
                  'u', 'v', 'w', 'x', 'y', 'z', '1',
                  '2', '3', '4', '5',
                  '6', '7', '8', '9', '0'};

char morse[36][6] = {".-", "-...", "-.-.", "-..", ".", "..-.", "--.",
                     "....", "..", ".---", "-.-", ".-..", "--", "-.",
                     "---", ".--.", "--.-", ".-.", "...", "-", "..-",
                     "...-", ".--", "-..-", "-.--", "--..",
                     "-----", ".----", "..---", "...--", "....-", ".....",
                     "-....", "--...", "---..", "----."};
```

Fonte: O próprio autor.

Figura 7 - Processo de Codificação

```
if (opcao == 1)  //codificar
{
    printf("Digite a palavra a ser codificada: ");
    scanf(" %[^\n]", palavra);

    tamanho = strlen(palavra);

    for(i=0;i<tamanho;i++)
    {
        for(j=0;j<36;j++)
        {
            if (tolower(palavra[i]) == letra[j])
            {
                fprintf(arquivo,"%s ", morse[j]);
                break;
            }
        }
    }
```

Fonte: O próprio autor.

2.2 Decodificação

Ainda utilizando nossos arrays para realizar a equivalência, o usuário, ao escolher a opção "Decodificar' no menu, deverá inserir o código morse que deseja ser decodificado. O programa irá armazená-lo e, após realizar uma verificação de que o termo inserido está em conformidade com as regras do código Morse e diferenciar as letras através dos espaços inseridos, percorrerá o item separando as letras individualmente e realizará a decodificação.

O processo é dado como encerrado quando o arquivo .txt com a decodificação é criado e fica acessível ao usuário.

Figura 8 - Processo de Decodificação

```
} else if(opcao == 2){    //decodificar
    printf("Morse a ser decodificado: ");
    scanf(" %[^\n]", palavra);
    tamanho = strlen(palavra);

    int pulador = 0;

    int x = 0;
    char *temporario = malloc(x);

    for (i = 0; i < tamanho + 1; i++)
    {
        if (palavra[i] == '-' || palavra[i] == '.')
        {
            x++;
            temporario = (char *)realloc(temporario, x);
            temporario[i - pulador] = palavra[i];
        }
        else if (palavra[i] == ' ' || palavra[i] == '\0')
        {
            pulador = i + 1;
            x++;
            temporario = (char *)realloc(temporario, x);
            temporario[x - 1] = '\0';

            for (j = 0; j < 36; j++)
            {
                if (strcmp(temporario, morse[j]) == 0)
                {
                    fprintf(arquivo,"%c", letra[j]);
                    break;
                }
            }
            x = 0;
            memset(temporario, 0, strlen(temporario));
        }
        else
        {
            printf("ERROR - MORSE SO PODE CONTER '.' '-' ' '\n");
            return 10;
        }
    }
    free(temporario);
```

Fonte: O próprio autor.

3 RESULTADOS E DISCUSSÕES

Após a finalização do código, foi checado o funcionamento da função codificadora e decodificadora. Ao inserir a palavra "IEEE", ".. . . . " é recebido como resultado e, ao repassar o mesmo código pela função decodificadora, "ieee" érecebido como resultado, assim, verificando o funcionamento correto do programa.

CONCLUSÕES

A partir da testagem do funcionamento do código foi possível aferir que tanto a codificação quanto a decodificação do Código Morse estavam funcionando e, logo, que a programação foi feita de maneira correta, assim, finalizando o projeto.

REFERÊNCIAS

Introdução à Programação em C. Disponível em:<https://www.inf.ufpr.br/roberto/ci067/01_intro.html>. Acesso em: 20 set. 2023.

Strings. Disponível em:<https://www.inf.ufpr.br/roberto/ci067/05_string.html>. Acesso em: 20 set. 2023.

Ponteiros. Disponível em:<https://www.inf.ufpr.br/roberto/ci067/08_pointers.html>. Acesso em: 20 set. 2023.

A Estrutura de Repetição for. Disponível em:<https://www.inf.ufpr.br/cursos/ci067/Docs/NotasAula/notas-15_Outras_Estruturas_Repeti.html#sec:forl>. Acesso em: 20 set. 2023.