

**UNIVERSIDADE FEDERAL DE VIÇOSA**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**

PEDRO CARDOSO DE CARVALHO MUNDIM - 3877

**FÓRUM TEMÁTICO 1 - SISTEMAS OPERACIONAIS (CCF 451)**

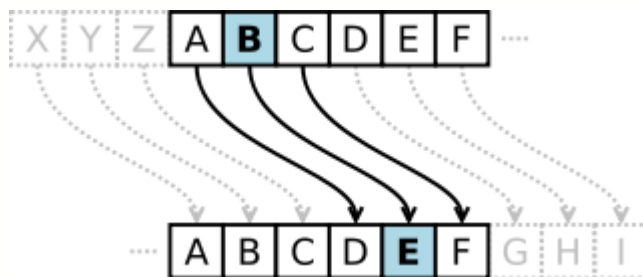
Primeiro Fórum Temático da disciplina Sistemas Operacionais - CCF 451, do curso de Ciência da Computação da Universidade Federal de Viçosa - Campus Florestal

Professor: Daniel Mendes Barbosa

FLORESTAL

2021

“Em criptografia, a **Cifra de César**, também conhecida como cifra de troca, código de César ou troca de César, é uma das mais simples e conhecidas técnicas de criptografia. É um tipo de cifra de substituição na qual cada letra do texto é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes.”



A ação de uma cifra de César é mover cada letra do alfabeto um número de vezes fixo abaixo no alfabeto. Este exemplo está com uma troca de três, então o B no texto normal se torna E no texto cifrado.

Fonte: [https://pt.wikipedia.org/wiki/Cifra\\_de\\_C%C3%A9sar](https://pt.wikipedia.org/wiki/Cifra_de_C%C3%A9sar)

Neste Fórum Temático foi desenvolvido um código que realiza a codificação de strings com base no conceito da Cifra de César, utilizando-se de duas threads, como foi especificado.

Vale ressaltar que a transformação será feita com o índice + 3, ou seja, a letra A, por exemplo, se tornará D, como mostrado na figura acima. Além disso, usei a tabela ASCII ao realizar o código. Será mostrado exemplos de execução após o código. A tabela ASCII pode ser encontrada neste link: <https://pt.wikipedia.org/wiki/ASCII>

Falando um pouco sobre as threads: a primeira irá verificar cada elemento da string (se é maiúsculo, minúsculo, pontuação, etc.); a segunda fará a conversão de cada um desses elementos com base no índice +3 explicado anteriormente para a string “cifrada” (para isso são apresentadas todas as condições dadas no código).

## Código

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Biblioteca para usar Threads
#include <pthread.h>

int i = 0;

// Define quando há alternância entre as threads
int definirAlternancia = 1;
int stringAuxiliar = 0;
char string[10000]; // String a ser codificada
char stringCifrada[10000]; // Resultado da codificação
int auxiliarDeslocamento = 0;
int valido = 1;

// Thread de verificação de cada elemento da String
void *verificaString(void *identificadorThread){
    // Passando caractere a caractere da string
    while(i < strlen(string)){
        if(definirAlternancia == 1){
            stringAuxiliar = string[i];
            /*
             Tabela ASC: Verificar se há somente letras
             minúsculas, maiúsculas números, espaço e
             pontuação
            */
            if(((stringAuxiliar >= 48 && stringAuxiliar <= 57)
            || (stringAuxiliar == 33) || (stringAuxiliar == 63)
            || (stringAuxiliar == 32) || (stringAuxiliar == 44)
            || (stringAuxiliar == 46) ||
            (stringAuxiliar >= 65 && stringAuxiliar <= 90) ||
            (stringAuxiliar >= 97 && stringAuxiliar <= 122)))){
                valido = 0;
            }
            // O caractere pode ser transformado?
            if(valido == 1){ // Pular caracteres especiais
                definirAlternancia = 1;
                ++i;
            }else{
                definirAlternancia = 0;
            }
        }
    }
}

// Thread que fará a conversão
void *aplicaCifra(void *identificadorThread){
    while(i < strlen(string)){
        if(definirAlternancia == 0){
            // Obtendo o valor ASC do respectivo char
            stringAuxiliar = string[i];
```

```

        if(stringAuxiliar == 32){ // Espaço vazio
            stringCifrada[i] = 32;
        }
        else if(stringAuxiliar == 44){ // Vírgula
            stringCifrada[i] = 44;
        }
        else if(stringAuxiliar == 46){ // Ponto final
            stringCifrada[i] = 46;
        }
        else if(stringAuxiliar == 33){ // Exclamação
            stringCifrada[i] = 33;
        }
        else if(stringAuxiliar == 63){ // Interrogação
            stringCifrada[i] = 63;
        }
        else{
            // Cifra de César deslocada 3 posições à direita
            auxiliarDeslocamento = stringAuxiliar + 3;

            // Criptografando as strings maiúsculas
            if(stringAuxiliar >= 65 && stringAuxiliar <= 90){
                if(auxiliarDeslocamento < 65){
                    auxiliarDeslocamento = stringAuxiliar + 24;
                }
            }

            // Criptografando números
            if(stringAuxiliar >= 48 && stringAuxiliar <= 57){
                if(auxiliarDeslocamento < 48){
                    auxiliarDeslocamento = stringAuxiliar + 8;
                }
            }

            // Criptografando as strings minúsculas
            else{
                if(auxiliarDeslocamento < 97 &&
                    auxiliarDeslocamento > 94){
                    auxiliarDeslocamento = stringAuxiliar + 24;
                }
            }

            // Armazenando o char codificado
            stringCifrada[i] = auxiliarDeslocamento;
        }
        ++i;

        /*
        Habilitando a Thread 1 para verificação do
        próximo elemento
        */

        definirAlternancia = 1;
    }
}
}

```

```
int main() {
    int j;
    printf("Digite uma string: ");
    scanf("%[^\n]s", string);
    setbuf(stdin, NULL);
    pthread_t Thread1, Thread2;

    // Criando as Threads
    pthread_create(&Thread1, NULL, verificaString, NULL);
    pthread_create(&Thread2, NULL, aplicaCifra, NULL);

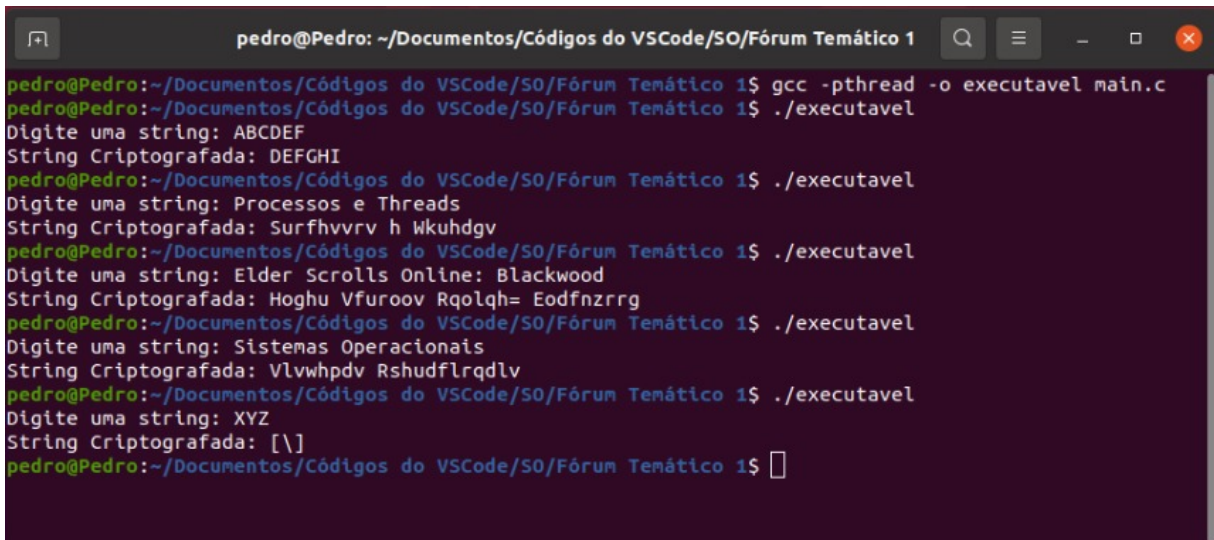
    // Esperando as Threads concluírem seus procedimentos
    pthread_join(Thread1, NULL);
    pthread_join(Thread2, NULL);

    printf("String Criptografada: ");

    for(j = 0; j < strlen(string); j++){
        printf("%c", stringCifrada[j]);
    }

    printf("\n");
}
```

## Algumas Execuções:



```
pedro@Pedro: ~/Documentos/Códigos do VSCode/SO/Fórum Temático 1
pedro@Pedro:~/Documentos/Códigos do VSCode/SO/Fórum Temático 1$ gcc -pthread -o executavel main.c
pedro@Pedro:~/Documentos/Códigos do VSCode/SO/Fórum Temático 1$ ./executavel
Digite uma string: ABCDEF
String Criptografada: DEFGHI
pedro@Pedro:~/Documentos/Códigos do VSCode/SO/Fórum Temático 1$ ./executavel
Digite uma string: Processos e Threads
String Criptografada: Surfhvrvv h Wkuhdgv
pedro@Pedro:~/Documentos/Códigos do VSCode/SO/Fórum Temático 1$ ./executavel
Digite uma string: Elder Scrolls Online: Blackwood
String Criptografada: Hoghu Vfuroov Rqolqh= Eodfnzrrg
pedro@Pedro:~/Documentos/Códigos do VSCode/SO/Fórum Temático 1$ ./executavel
Digite uma string: Sistemas Operacionais
String Criptografada: Vlvwhpdv Rshudflrqdlv
pedro@Pedro:~/Documentos/Códigos do VSCode/SO/Fórum Temático 1$ ./executavel
Digite uma string: XYZ
String Criptografada: [\]
pedro@Pedro:~/Documentos/Códigos do VSCode/SO/Fórum Temático 1$
```

## Observações Sobre as Execuções e o Código:

- A primeira execução é o exemplo mostrado na primeira figura.
- A segunda, a terceira e a quarta execuções mostram que o código considera **espaços em branco**, bem como o uso dos **dois pontos** na terceira execução (que se transformou no símbolo de igualdade). Como descrito no código ele também considera as outras pontuações (., ! ? :). Todas estas considerações estão dadas nas condicionais do código.
- A última demonstra um exemplo em que **XYZ** se transformam em **[ \ ]**. Ao observar a tabela ASCII e somar +3 ao valor de cada um desses caracteres chegaremos em tal transformação.
- Também é possível ver que o código faz a diferenciação entre letras **maiúsculas** e **minúsculas**.
- Não foram utilizados **números** nos exemplos, mas como mostrado no código, também serão considerados.

## Referências Utilizadas

- [1] [https://pt.wikipedia.org/wiki/Cifra\\_de\\_C%C3%A9sar](https://pt.wikipedia.org/wiki/Cifra_de_C%C3%A9sar)
- [2] <https://pt.wikipedia.org/wiki/ASCII>
- [3] <https://www.geeksforgeeks.org/thread-functions-in-c-c/>