

**Centro Universitário**

**Senac**

**Alunos:**

**Higor Viana de Morais**

**Josué Batista Matos Deschamps de Melo**

**Ygor Kaique de Souza Pinto**

**Relatório do EP: Fuse e Esteganografia**

**Professor: Eduardo Heredia**

**Curso: Bacharelado em Ciência da Computação**

**Disciplina: Sistemas Operacionais**

**Instituição: Centro Universitário Senac Santo Amaro**

**São Paulo**

**2020**

## Introdução

Neste projeto o desafio proposto era de se realizar o uso de Fuse (Filesystem in Userspace) e Esteganografia, que é uma forma de segurança por obscurantismo implementado com a linguagem C.

Realizamos um estudo no Github de Sadimanna para entender como funciona a manipulação de imagens em Bitmap de 16 cores, ajudando na implementação da esteganografia, e ainda com as referências que o Professor Eduardo Heredia passou durante o semestre.

Para que seja possível a manipulação de imagens, usamos imagens de Bitmap de 16 cores para conseguir fazer a manipulação do arquivo, com o uso de arquivo em txt para ser colocado dentro da imagem.

Em nosso projeto utilizamos o sistema operacional Ubuntu na versão 20.04.1, o Git para fazer a interação do grupo e o VSCode como editor de texto para programar o código e executar todos os procedimentos envolvidos.

## Desenvolvimento

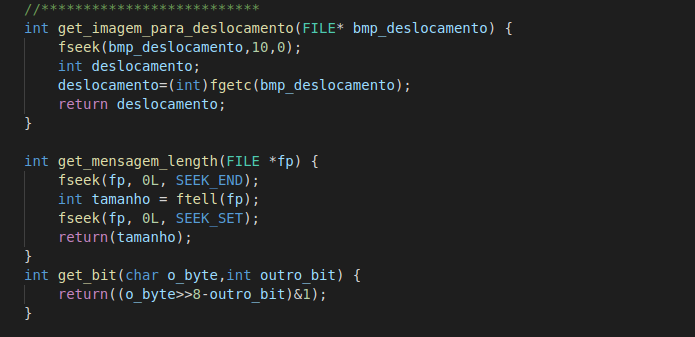
Iniciamos o projeto com a parte da esteganografia, pois tivemos mais facilidade de implementar. Na sequência consultamos o material de aula e fizemos pesquisas relacionadas ao FUSE. Porém tivemos dificuldade para juntar parte da esteganografia com a parte do FUSE.

Explicando algumas funções:

get\_imagem\_para\_deslocamento(): Captura a imagem para deslocamento e faz uma conversão para inteiro, fazendo ser possível o uso do binário

get\_mensagem\_length(): Configura o tamanho da mensagem.

get\_bit(): Captura cada bit da imagem para leitura dos bits.



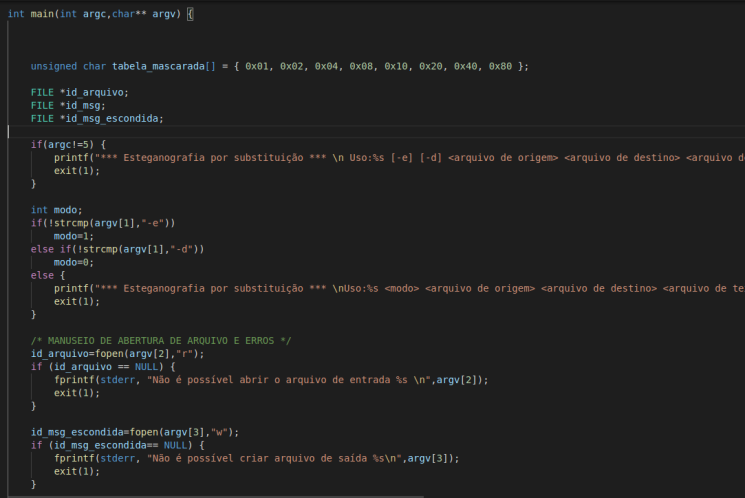
tabela\_mascara[]: Serve para converter a imagem em binário e fazer a esteganografia.

id\_arquivo: Identificador do Arquivo, para saber qual arquivo está sendo manipulado durante a execução do código.

id\_msg: Identificador da mensagem que está sendo passada para se tornar uma mensagem escondida.

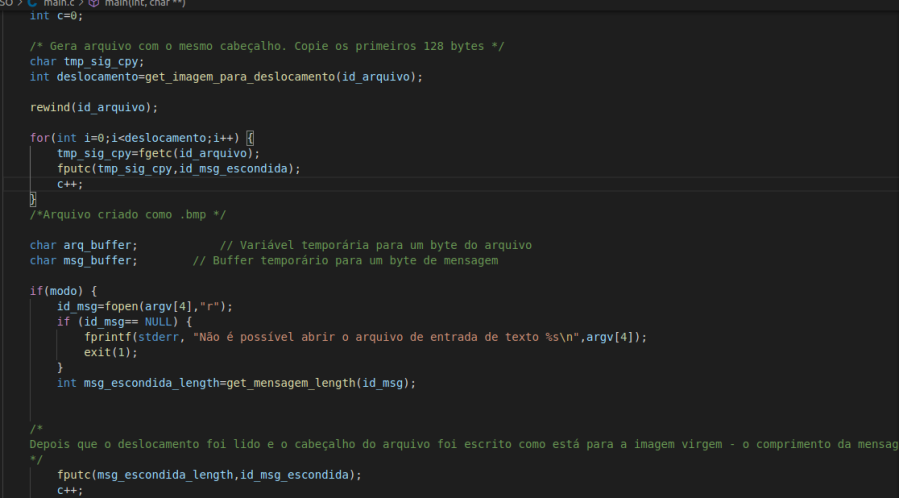
id\_msg\_escondida: Identificador da mensagem que foi escondida.

Algumas passagens de argumentos são tratadas com if para assim fazer uma espécie de contorno com possíveis exceções na execução da aplicação.



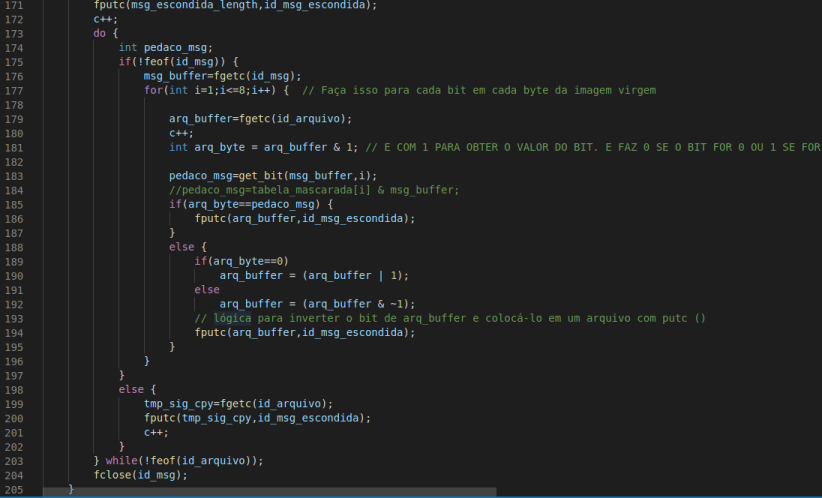
arq\_buffer: variável temporária para um byte do arquivo

msg\_buffer: buffer temporário para um byte de mensagem



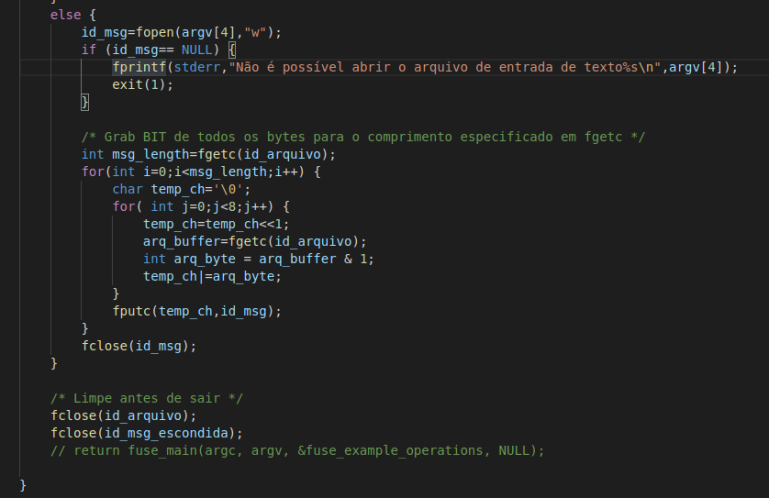
O “for” abaixo fica responsável por fazer uma varredura pela imagem e conseguindo assim obter cada bit de cada byte.

Com o uso da variável pedaco\_msg, que captura cada caractere da mensagem que será passada e da variável arq\_buffer que será comparada com o pedaco\_msg.

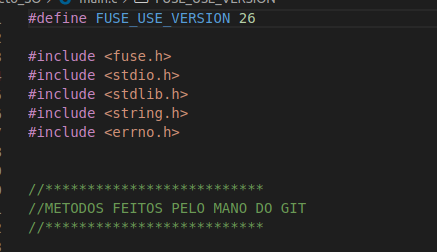
Caso as variáveis sejam iguais, a função fputc(), função essa que escreve o caractere c (convertido para unsigned char) no fluxo de saída será chamada. 

Neste segundo “for” temos todos os procedimentos para que possam ser juntados os bits e bytes da imagem passada anteriormente.

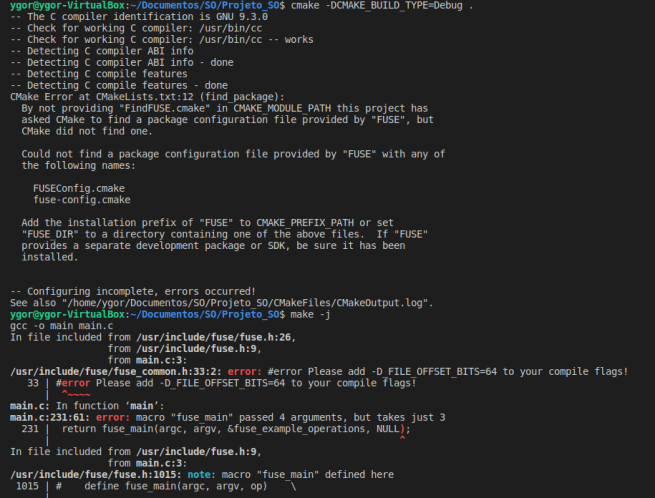
Após isso, os arquivos são fechados para que não fiquem como processo sendo utilizado na memória.

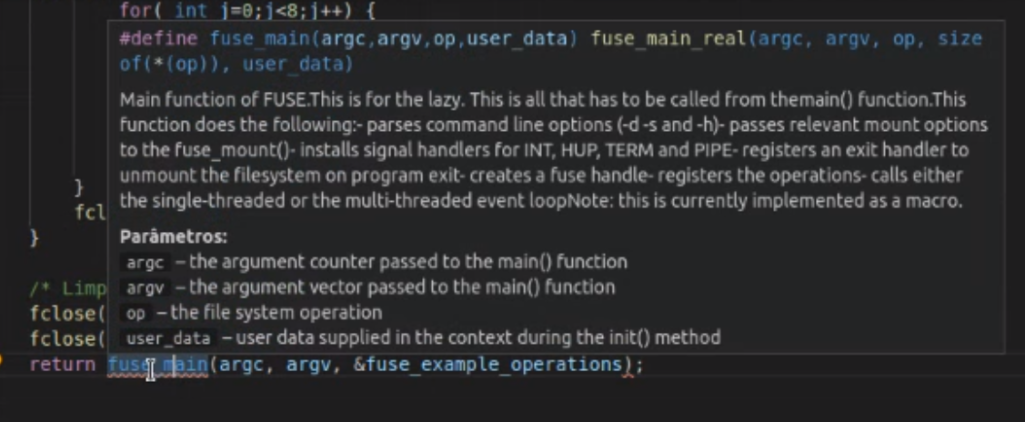


Abaixo foram utilizadas as seguintes bibliotecas:



Ao tentar executar o código usando o FUSE, tivemos os seguintes erros:





## Conclusão

Conseguimos atingir boa parte do que foi proposto, já que não foi possível executar os comandos necessários para a execução da parte do FUSE, porém, conseguimos implementar a esteganografia e entender de uma maneira melhor como é feita a esteganografia de arquivos.

## Referências

**Documentação do fuse** - [libfuse/libfuse: The reference implementation of the Linux FUSE (Filesystem in Userspace) interface (github.com)](https://github.com/libfuse/libfuse)

**URFJ** - <https://www.gta.ufrj.br/ensino/eel878/redes1-2016-1/16_1/esteganografia/>

**Wikipedia -** [**https://en.wikipedia.org/wiki/Filesystem\_in\_Userspace**](https://en.wikipedia.org/wiki/Filesystem_in_Userspace)

**Engineering -** [**https://engineering.facile.it/blog/eng/write-filesystem-fuse/**](https://engineering.facile.it/blog/eng/write-filesystem-fuse/)

**GitHub sobre leitura de arquivos BMP - Sadimanna -** [**https://github.com/sadimanna/compression/blob/master/readbmp.c**](https://github.com/sadimanna/compression/blob/master/readbmp.c)