
# Ferramenta para modificar permissões de ficheiros

**XMOD**

Sistemas Operativos

Grupo 1 – Turma 3

Mestrado Integrado em Engenharia Informática e Computação


| | |
|---|---|
| Angela Cruz | up201806781 |
| Eduardo | |
| Filipe Pinto | up201907747 |
| Luísa Maria Mesquita | up201704805 |
| Pedro Moreira | up201904642 |

# Índice

**Índice** 1

**1. Introdução** 2

**2. Estruturas de dados e variáveis** 2

**3. Funções** 3

**4. Compilação e execução** 4

**4.1. Compilação** 4

**4.2. Execução** 4

**5. Conclusão** 4

# 1. Introdução

O projeto teve como objetivo a criação de um comando semelhante ao *chmod* (usado para a modificação de permissões de ficheiros).

O projeto foi estruturado da seguinte forma:

1. Medição dos tempos de execução.
2. Criação de uma estrutura e inicialização dos seus campos com informações sobre os argumentos introduzidos na linha de comandos.
3. Obtenção das permissões atuais de um ficheiro, cálculo das permissões novas e alteração destas.
4. Criação de processo-filhos sempre que for encontrado um subdiretório e uso da chamada execve como criação duma nova instância do programa. (opção "-R")
5. Implementação das opções "-v" e "-c".
6. Verificação de erros ao longo do programa.
7. Tratamento do ficheiro definido pela variável de ambiente LOG_FILENAME.
8. Tratamento do sinal SIGINT e apresentação da informação textual pedida na saída padrão.

# 2. Estruturas de dados e variáveis

A estrutura principal do programa, *Arguments*, guarda todas as informações introduzidas na linha de comandos e que são usadas depois ao longo do programa. A inicialização dos campos da estrutura é feita na função *InitializeArguments*().

```
struct Arguments
{
    int mode_octal;      //mode introduced in octal
    char *mode;          //mode introduced (non octal)
    bool mode_is_octal; //true if mode is in octal; false otherwise
    char *path_name;     //path with the directory/file
    bool is_dir;         //true is path_name is a directory
    bool option_v;       //true if mode '-v' was introduced
    bool option_c;       //true if mode '-c' was introduced
    bool option_R;       //true if mode '-R' was introduced
};
```

# 3. Funções

O programa chama várias funções e inicializa várias variáveis no main do programa. Por ordem, o flow do programa dá-se da seguinte forma:

1. O programa começa por "armadilhar" os sinais, isto é, atribuir handlers que indicam como manipular a sua receção para que não ajam por defeito. O *handler* usado para este efeito é o *SignalFunc*() que manipula o sinal CTRL + C enviado pelo utilizador. Esta função manipula este sinal e apresenta os processos que estão em execução, o diretório atual, o número de ficheiros encontrados até ao momento e o número de ficheiros modificados. No final, pergunta ao utilizador se deseja terminar ou continuar a execução do programa. Mediante a resposta, o programa age em conformidade.
2. De seguida, inicializa variáveis que permitem contar o tempo de execução do programa.
3. Depois, executa a função *InitializeArguments*() que inicializa os argumentos para a estrutura *Arguments* e trata dos erros relativos a estes, usando as funções *PrintError*() e *PrintManual*().
4. Caso a opção "-R" não tenha sido introduzida, o programa limita-se a alterar as permissões do ficheiro/directório passado na linha de comandos, chamando a função *ChangePermissions*().
5. Por outro lado, caso a opção "-R" tenha sido introduzida, o programa executa a função *ProcessRecursive*(), que percorrerá todos os ficheiros e subdirectórios que encontre no *path* introduzido. Altera as permissões tanto do ficheiro como dos subdiretórios. Encontrando um subdiretório, cria um novo processo-filho que executará a função *execve*(). O *execve*() cria uma nova instância para cada subdiretório encontrado. Assim, se o ficheiro a testar tem 3 subdiretórios, são criados 3 processos-filhos.

Outras funções auxiliares foram criadas e é possível ver mais informações sobre estas no ficheiro "*xmod.h*".

# 4. Compilação e execução

## 4.1. Compilação

```
make clean
make
```

ou

```
gcc -Wall -o xmod.o xmod.c
```

## 4.2. Execução

Para executar, escrever "./xmod.o" e os respectivos argumentos. Com a seguinte linha, é possível ver mais informações de execução.

```
./xmod.o --help
```

# 5. Conclusão

De seguida apresentamos as tarefas mencionadas na introdução que cada um fez e/ou contribuiu.

- Angela Cruz: Tarefas 2, 3 e 6
- Eduardo
- Filipe Pinto: Tarefas 1, 7 e 8
- Luísa Maria Mesquita: Tarefas 2, 4, 6 e 7
- Pedro Moreira: Tarefa 3 e 5