



# **Ferramenta para modificar permissões de ficheiros**

## **Relatório de projeto**

### **Sistemas Operativos**

Mestrado Integrado em Engenharia Informática e Computação (MIEIC)  
2ºAno 2ºSemestre - 2020/2021

Turma 7 - Grupo 4:

Ana Matilde Barra up201904795  
Ângela Coelho up201907549  
Nuno Castro up202003324  
Patrícia Oliveira up201905427

## Índice

1 – Introdução	<b>2</b>
2 – Detalhes de funcionalidades implementadas	<b>3</b>
2.1 Métodos de invocação	3
2.2 Geração de registos de execução	4
2.3 Tratamento de sinais	5
3 – Estruturação do código	<b>6</b>
4 – Autoavaliação	<b>6</b>

# 1 – Introdução

Este projeto surge no âmbito da unidade curricular de Sistemas Operativos em que nos foi proposto a elaboração de uma ferramenta de modificação de permissões de acesso ficheiros e diretórios com o objetivo de pôr em prática programação de interação com sistemas operativos, neste caso, o Unix/Linux.

Em Unix, as permissões dividem-se em tipo de acesso e a classe de utilizador a quem será permitido o acesso. Os tipos de acesso que um ficheiro pode ter podem ser leitura (**r**), escrita (**w**) e execução (**x**) enquanto as classes de utilizador podem ser o dono de ficheiro (**u**), o grupo ao qual o utilizador pertence (**g**) ou todos os outros utilizadores (**o**).

A ferramenta implementada tem como base o comando **chmod** cuja funcionalidade é a mesma pretendida para a nossa ferramenta, com algumas adaptações requeridas pelo enunciado do projeto.

## 2 – Detalhes de funcionalidades implementadas

### 2.1 Métodos de invocação

A nossa ferramenta suporta dois tipos de invocação:

- **xmod [options] mode file/dir;**
- **xmod [options] octal-mode file/dir.**

Em que:

- **options** pode ser:

- -v, o modo verboso que apresenta informação acerca do ficheiro;
- -c, semelhante ao modo verboso mas só apresenta a informação no caso de haver modificações;
- -R, o modo recursivo em que o diretório selecionado é percorrido e processado recursivamente.

- **file/dir** é o nome do ficheiro ou diretório a processar;

- **mode** tem o seguinte formato '[ugoa](-+)(rwx)' onde:

- [ugoa] referem-se às classes de utilizador mencionadas anteriormente mais a adição de 'a', que aplica a todas as classes;
- (-+)= referem-se à remoção, adição ou substituição de permissões, respetivamente;
- (rwx) referem-se aos tipos de acesso mencionados anteriormente.

- **octal-mode** é uma sequência de três ou quatro números referente à representação octal das permissões. No caso de serem introduzidos 4 números, o primeiro deverá ser 0, este, porém, pode ser omitido (ex: 0753 = rwxr-x-wx).

De reforçar que a ordem dos argumentos de entrada é respeitada escrupulosamente e não são permitidas repetições do mesmo tipo de acesso de permissões. É de notar que também não é permitido repetir as opções, no entanto é possível colocá-las de qualquer ordem.

## 2.2 Geração de registos de execução

De modo a facilitar a análise ao funcionamento da nossa ferramenta foi concebida a funcionalidade de registo de eventos de execução.

Este registo é efetuado apenas em execuções que se realizem após a definição da variável **LOG\_FILENAME** pelo utilizador. Esta variável determina o nome do ficheiro onde irá ser efetuado o registo dos eventos, um por linha no respeitando o seguinte formato:

**instant ; pid ; event ; info**

Em que:

- **instant** refere-se ao instante de tempo após o início da execução do programa da ocorrência do evento;
- **pid** refere-se ao identificador do processo no qual ocorre o evento;
- **event** refere-se ao respectivo tipo de evento;
- **info** é a informação referente ao tipo de evento.

Para este registo foram considerados os seguintes tipos de eventos:

- *PROC\_CREAT* - criação de processo;
- *PROC\_EXIT* - fim de processo;
- *SIGNAL\_RECV* - sinal recebido pelo processo;
- *SIGNAL\_SENT* - sinal enviado por processo;
- *FILE\_MODF* - alteração às permissões de um ficheiro ou diretório por um processo.

A cada tipo de evento está associada a respetiva informação específica sobre o mesmo. No entanto, com uma pequena adaptação, na situação de um sinal ser enviado para todos os processos de um determinado grupo, onde o identificador desse grupo é indicado entre parênteses curvos.

## 2.3 Tratamento de sinais

Sempre que é enviado o sinal **SIGINT** durante a execução do programa, este é processado apenas pelo processo principal. No caso de não existirem processos descendentes por terminar é enviado o sinal **SIGUSR2** ao processo principal enquanto no caso contrário é enviado o sinal **SIGUSR1** a todos eles, incluindo o processo principal. Este último é ignorado pelo processo principal enquanto todos os descendentes não terminarem de fazer display da sua informação na linha de comandos com o seguinte formato:

**pid ; fich/dir ; nftot ; nfmod**

em que:

- **pid** é o identificador de processo
- **fich/dir** é o nome do ficheiro ou diretório
- **nftot** é o número total de ficheiros encontrados nesse processo
- **nfmod** é o número de ficheiros modificados por esse processo

O subdiretório a iterar para o qual um novo processo é criado pertence à contagem do número total de ficheiros encontrados e modificados do mesmo.

Findado o display da sua informação, cada descendente envia o sinal **SIGUSR2** para o seu processo parente a informar do término e **SIGSTOP** a si próprio.

Quando o processo principal recebe **SIGUSR2** este dá display da sua informação e é perguntado ao utilizador se este realmente pretende terminar o programa. Consoante a resposta introduzida ou todos os processos são terminados ou retomados.

### 3 – Estruturação do código

O código foi organizado por módulos. Cada módulo tem a sua responsabilidade para o funcionamento da ferramenta. Repartindo as tarefas por módulos torna-se mais fácil a manutenção e gestão de todo o projeto, bem como a distribuição das mesmas pelos diversos elementos do grupo.

Todos os módulos definidos e suas funções estão descritos na seguinte tabela:

<b>main</b>	Módulo principal para a execução da ferramenta.
<b>xmod</b>	Neste módulo encontram-se funções para modificação de permissões.
<b>inputcheck</b>	Este módulo é responsável pelo processamento do input proveniente da linha de comandos para posterior utilização pelos restantes módulos.
<b>filesystem</b>	Neste módulo encontram-se definidas as funções destinadas à navegação recursiva de diretórios.
<b>signals</b>	Neste módulo estão definidas funções para processamento de sinais.
<b>logger</b>	Neste módulo foram definidas funções para manipulação do ficheiro de registo de eventos.
<b>utils</b>	Neste módulo encontram-se funções gerais de auxílio aos restantes módulos.

### 4 – Autoavaliação

Abaixo apresentamos a divisão percentual (aproximada) da contribuição de cada membro do nosso grupo para o resultado final deste projeto:

- Ana Matilde Barra (up201904795) - 25%
- Ângela Coelho (up201907549) - 25%
- Nuno Castro (up202003324) - 25%
- Patrícia Oliveira (up201905427) - 25%

Consideramos que todos os elementos do grupo trabalharam igualmente no trabalho e contribuíram para o seu desenvolvimento.