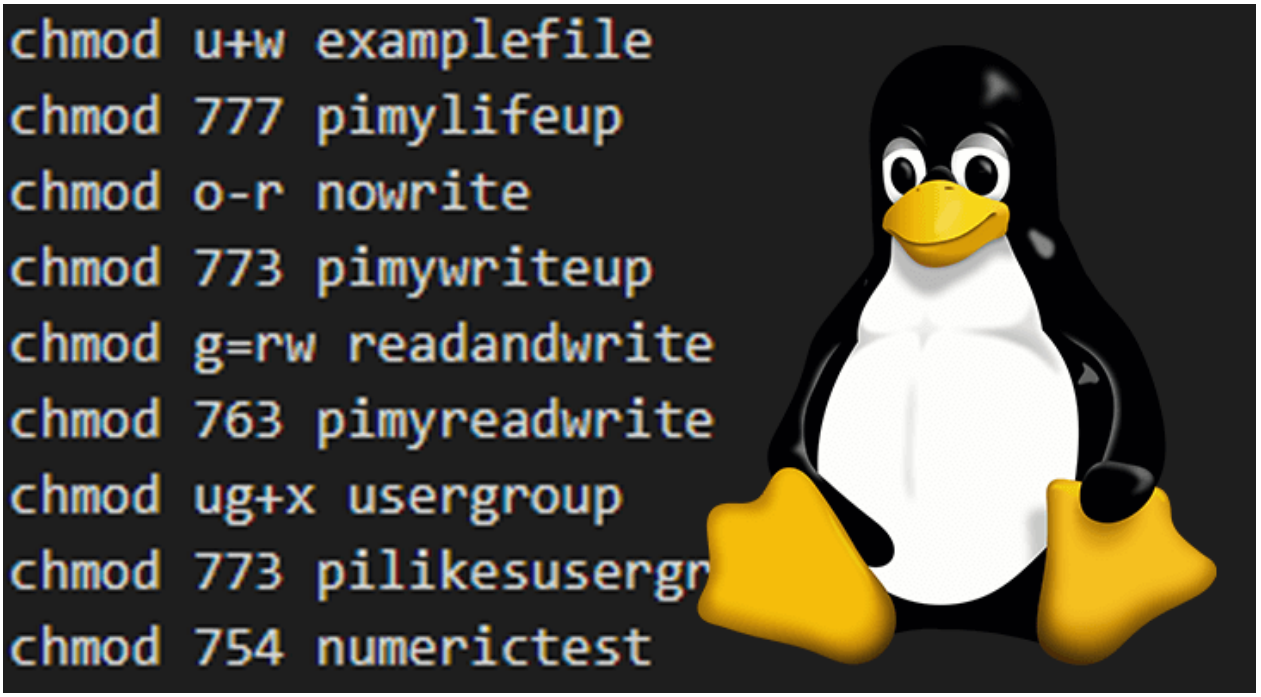
U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

Sistemas Operativos



MP1: xmod, ferramenta para modificar permissões de ficheiros

Marcelo Henriques Couto	up201906086@fe.up.pt
Miguel Azevedo Lopes	up201704590@fe.up.pt
Pedro Miguel Jesus da Silva	up201907523@fe.up.pt
Sofia Ariana Moutinho Coimbra Germer	up201907461@fe.up.pt

Introdução

Para o desenvolvimento do MP1 foi nos pedido para desenvolver a ferramenta xmod, devendo usar como referência o comando chmod (change filemode bits), que permite modificar permissões de acesso a ficheiros e diretórios.

Para isto teve-se que fazer a leitura dos argumentos e parse da informação, alteração das permissões com base nestes, implementação de funcionalidades referentes às opções, configurar tratamento de sinais, fazer registos de execução, etc.

Requisitos Funcionais

Os requisitos funcionais pedidos foram cumpridos com sucesso. No programa as três opções pedidas, verboso (-v), verboso apenas em modificações (-c) e recursivo (-r) funcionam como é pedido e de acordo com o comando original, podendo aparecer em qualquer posição nos argumentos. Quanto aos dois modos de funcionamento pedidos, também estão programados para serem fiéis ao chmod, sendo que no modo não numérico a ordem das permissões escolhidas não precisa de ser exatamente rwx, como é pretendido. A geração de registos de execução e o tratamento de sinais encontra-se igualmente implementado como é pedido.

Detalhes de implementação

O parse das opções e do modo começam na função parse() que percorre todos os argumentos passados até os encontrar para depois serem processados nas devidas funções.

O módulo que trata dos modos quase na sua integridade é o modes, que recebe as novas permissões, o grupo às quais elas vão ser alteradas e lê as permissões atuais de um dado ficheiro, sendo isto importantes para o caso de manter uma das permissões que já estava e que não foi pedida para ser alterada. Através de operações bitwise entre o modo antigo e o modo novo, será possível construir o resultante para depois ser usado na chamada ao sistema chmod().

As opções são processadas e guardadas numa struct global. A função que o faz encontra-se no módulo utils, um módulo que está destinado a funcionalidades diversas e que complementam os outros módulos. Depois de guardadas as opções, o seu efeito far-se-á notar na função executer.
Na implementação do modo recursivo, caso o ficheiro a analisar fosse novamente um diretório é usado um fork e execvp de modo a correr o programa de novo, na sua integridade, como um novo processo.

O módulo logfile é responsável pela escrita dos registos de excecução. Uma função initRegister() é chamada apenas no processo inicial (isto é conseguido através do recurso a variáveis de ambiente) e além de detetar o estado da variável de ambiente que guarda o path do ficheiro de registo, trunca-o. A escrita dos registos é feita por múltiplas funções: uma principal, que recebe evento e informação como argumentos e outras, específicas a cada evento, que formatam a informação consoante a situação e chamam a primeira.

Em relação aos sinais e ao módulo signal_handling: foram definidos handlers para múltiplos sinais. O envio do sinal SIGINT e a resposta adequada ao mesmo através de dois handlers, um para o processo inicial, e outro para o resto. Cada handler vai imprimir a informação pretendida no

terminal. O processo inicial vai perguntar ao utilizador se deseja continuar e, consoante a resposta, enviar um sinal SIGCONT ou SIGTERM aos outros processos. O primeiro força os outros processos a saírem da pausa auto-induzida por auto-envio do sinal SIGTSTP. O segundo vai obrigá-los a terminar.

Autoavaliação

Marcelo Couto	30 %
Miguel Lopes	30 %
Sofia Germer	30 %
Pedro Silva	10 %