

1º Mini Projeto:

Ferramenta para modificar permissões de ficheiros

Objetivos

Familiarização com a programação de interação com um sistema operativo (aqui, Unix/Linux), envolvendo processos, intercomunicação por sinais e acesso ao sistema de ficheiros.

Descrição Geral

A ferramenta a desenvolver, denominada **xmod**, deve ter como referência o comando **chmod** (*change file mode bits*), que permite modificar permissões de acesso a ficheiros e diretórios.

Em Unix, tais permissões são função do tipo de acesso e da classe de utilizador onde se insere quem pretende o acesso. As permissões estão associadas em três conjuntos, cada um dos quais associado ao dono do ficheiro (**u**), ao grupo a que o utilizador pertence (**g**) e a todos os outros utilizadores (**o**); cada conjunto contempla três tipos de acesso: de leitura (**r**), escrita (**w**) e execução (**x**).

A representação deste conjunto de permissões dá-se em duas maneiras:

- indicação em linha das letras **r w x**, posicionadas nesta ordem e sequencialmente para cada um dos “tipos de utilizador” **u g o**, nesta ordem. No total há uma sequência de 9 caracteres e se não houver um qualquer tipo de permissão para algum tipo de utilizador, um ‘-’ substitui a letra em falta; por exemplo, **rwxr-xr-x**, onde só o dono do ficheiro tem permissão de escrita;
- indicação de uma sequência do dígito ‘0’ mais 3 dígitos em base octal, estando cada um destes associado a cada um dos tipos de utilizador na ordem **u g o** e, numa interpretação posicional binária, **r w x** correspondem a um bit que será 1 se houver essa permissão ou 0 na sua ausência; por exemplo, **0755**, onde só o dono do ficheiro tem permissão de escrita.

xmod deve reproduzir com rigor um subconjunto das funcionalidades de **chmod**, que incluem a maneira como “*symbolic links*” são tratados e o formato da informação apresentada no final da execução.

As funcionalidades de **xmod** relativas à linha de comando estão listadas abaixo, na secção Requisitos Funcionais. A listagem e descrição completas das funcionalidades de **chmod** podem ser vistas com “*man chmod*”.

Requisitos Funcionais

xmod deverá suportar os seguintes métodos de invocação:

xmod [OPTIONS] MODE FILE/DIR

xmod [OPTIONS] OCTAL-MODE FILE/DIR

onde:

FILE/DIR é o nome inequívoco (absoluto ou relativo) de um ficheiro ou diretório a processar, podendo ser um *link* simbólico.

OPTIONS poderá ser um ou mais de:

- **-v** : modo verboso, que apresenta muita informação sobre todos os ficheiros encontrados;

- **-C** : semelhante a modo verboso, mas apenas apresentando informação quando é efetuada uma modificação;
- **-R** : o directório indicado em **FILE/DIR** é processado e atravessado recursivamente, processando-se toda a árvore de directórios abaixo.

MODE possui o formato '[ugoa](-+)(rwx)+', indicando:

- **[ugoa]** : opcionalmente, o tipo de utilizador a ser afetado, **u** **g** ou **o**, ou todos os tipos, **a** (de *all*); por omissão, assume-se 'a', mas os bits assinalados em "*umask*" (*file mode creation mask*) não serão alterados;
- **(-+)** : que as permissões (imediatamente a seguir) serão, respectivamente, removidas, adicionadas ou substituirão as que já existem, removendo as não mencionadas;
- **(rwx)** : as permissões a serem modificadas, podendo várias ser indicadas.

OCTAL -MODE é uma sequência de 4 números, na representação octal de permissões, já apresentada atrás.

Funcionalidades Adicionais

Geração de registos de execução

Para facilitar a análise, desenvolvimento e avaliação da ferramenta **xmod**, devem ser registados num ficheiro a ocorrência de diversos eventos, listados mais abaixo (por exemplo, criação de processos). Os registos devem ser efectuados se a variável de ambiente **LOG_FILENAME** for definida pelo utilizador; nesse caso, ela determina o nome do ficheiro de registo. Se tal ficheiro já existir, deve ser truncado (e escrito como se fosse um novo); se não existir, deve ser criado.

Qualquer um dos processos participantes na operação do programa acede ao ficheiro, acrescentando-lhe informação, linha a linha, no seguinte formato:

instant ; pid ; action ; info

onde:

- **instant** é o instante de tempo imediatamente anterior ao registo, medido em milissegundos, tendo como referência o instante em que o programa começou a executar;
- **pid** é o identificador do processo que faz o registo da linha;
- **event** é a identificação do tipo de evento que afectou o processo (ver já a seguir);
- **info** é informação adicional associada ao evento (ver já a seguir).

A tabela seguinte mostra a representação dos tipos de eventos que devem ser registados e a correspondente informação adicional:

	event	info
criação de processo	PROC_CREAT	argumentos da linha de comandos que origina o processo
terminação de processo	PROC_EXIT	código de saída do processo (<i>exit status</i>)
recepção de sinal por processo	SIGNAL_RECV	sinal recebido pelo processo (e.g., SIGINT)
envio de sinal por processo	SIGNAL_SENT	sinal enviado, seguido do PID do processo alvo, no formato: signal : pid (e.g. , SIGINT : 987)

	event	info
modificação de permissões de ficheiro /directório por processo	FILE_MODF	nome canónico do ficheiro/directório que teve as permissões alteradas, seguido dos valores em octal das permissões anteriores e novas no formato: fich/dir : oldPerm : newPerm (e.g., /tmp/fich12 : 0666 : 0644)

Tratamento de sinais

Estando **xmod** em execução, quando lhe for enviado o sinal **SIGINT** (normalmente originado na consola por CTRL-C), deverá ser apresentado na saída padrão um conjunto de linhas com informação textual referente aos processos em execução. Cada linha, para cada processo, terá o formato:

pid ; fich/dir ; nftot ; nfmod

onde

- **pid** é o identificador do processo;
- **fich/dir** é o nome canónico do ficheiro ou directório especificado em **FILE/DIR** da linha de comando que originou o processo (ver, abaixo, Requisitos Arquitecturais);
- **nftot** é o número de ficheiros encontrados até ao momento;
- **nfmod** é número de ficheiros já modificados.

No final das linhas, deve ser apresentada uma mensagem perguntando ao utilizador se quer terminar o programa ou se quer que a execução prossiga. Recebida a resposta, o programa age em conformidade.

Requisitos Arquitecturais

Apesar da estrutura do programa ser deixada a cargo de cada grupo de trabalho, há um conjunto de requisitos arquitecturais que são exigidos:

- quando **xmod** é iniciado pelo utilizador, o primeiro processo a ser executado (processo inicial) deve, antes de terminar, aguardar pela terminação de todos os restantes processos do programa;
- sempre que **xmod** é invocado com a opção **-R**, cada um dos eventuais processos criados deve analisar somente o directório que lhe for passado por **FILE/DIR** e criar um (sub)processo por cada um dos sub-directórios que eventualmente possam existir; cada um desses (sub)processos executará uma nova instância de **xmod**, com as mesmas opções da invocação feita pelo utilizador, com a excepção do parâmetro **FILE/DIR** que terá a identificação do sub-directório que motivou a sua criação; e assim sucessivamente.

Plano de Trabalho

Sugere-se que o mini-projecto, que produzirá a ferramenta **xmod**, seja desenvolvido por etapas:

1. experimentação de múltiplas funções/chamadas ao sistema que irão ser úteis ao **xmod**;
2. escrita de uma primeira versão do **xmod** que trabalhe apenas com um ficheiro ou com um directório, que faça registos e não utilize sinais;
3. adição do tratamento de sinais à 1a. versão de **xmod**, formando uma segunda versão;
4. alargamento da 2a. versão de **xmod** ao caso recursivo (processamento de toda uma árvore de directórios), mas sem grande preocupação com a boa funcionalidade dos sinais: será a terceira versão;
5. afinação da 3a. versão de **xmod**: sinais, registos, etc. produzindo, assim, a versão final.

Notas Finais

- Os nomes do programa e da variável de ambiente, os formatos dos argumentos da linha de comando e das linhas a apresentar na saída padrão e no ficheiro de registos e todos os outros aspectos que foram especificados devem ser respeitados escrupulosamente.
- Tudo o que não tiver sido especificado deverá ser implementado usando como referência o comando **chmod**.

Entrega do Trabalho

Deverá ser feita de acordo com as instruções comuns a todos os mini-projectos, disponíveis na área Moodle da disciplina, e que também incluem as datas de submissão.

Avaliação

A avaliação funcional do programa **xmod** será feita no **GNOMO.FE.UP.PT** e da forma o mais automática possível. Pormenores dos testes de verificação (e de avaliação) dos programas submetidos pelos grupos serão publicados muito em breve.