

Clarificação de algumas dúvidas relativamente ao enunciado do MP1 colcadas por um colega vosso (em nome do grupo).

Perguntas	Respostas
PARSING	
<p>1) Os métodos de invocação de xmod estão especificados como</p> <pre> ''' xmod [OPTIONS] MODE FILE/DIR xmod [OPTIONS] OCTAL-MODE FILE/DIR ''' O '[OPTIONS]' no início sugere que as opções devem ser dadas antes dos restantes argumentos. Porém, esta mesma notação é usada na maior parte dos programas linux e mesmo assim eles permitem que as opções apareçam em qualquer posição, desde que antes de um '--' Por exemplo, um dos métodos de invocação de chmod é `chmod [OPTION]... MODE[,MODE]... FILE...`, mas é válido fazer `chmod 0777 nome_ficheiro -v` ou até `chmod 0777 -v nome_ficheiro`. Por causa disso, estamos em dúvida se o objetivo era permitir em xmod também que as opções estejam em qualquer ordem (desde que antes de um '--'), ou se para simplificar apenas devem estar antes dos restantes argumentos.</pre>	<p>1) É verdade q mtos programas Unix aceitam opções em qq ordem; mas tb é verdade q mtos não aceitam. De qq modo, como foi pedido, cumpram escrupulosamente a especificação: no xmod, indiquem as opções pela ordem indicada.</p>
<p>2) No enunciado alterado está escrito:</p> <pre> `<rxw>` : indica qual ou quais permissões (até três!) devem ser modificadas. Devido à versão anterior do enunciado, não tínhamos essa restrição (ou seja, tal como o programa `chmod`, aceitávamos coisas como `-rwxr`). Passando para a nova versão do enunciado, queríamos confirmar se este limite de três permissões: - Permite ter permissões repetidas (ex. `-rrr` ou `-rwr` ou `-ww`) - Permite ter permissões numa ordem diferente do convencional (ex. `-xr` ou `-wrx`) </pre>	<p>2) a alteração do enunciado foi por causa de simplificar um trabalho de programação q n era essencial à disciplina. Assim, em cada invocação de xmod só podem escolher uma opção de entre as 'u g o a'; por isso só faz sentido escolherem (em cada invocação) r ou w ou x ou combinações deles, sem repetições: q significaria 'rrr' para 'u'? O '-' (ausência de permissão) tb n está previsto na escolha: <rxw>.</p> <p>Pedro Gonçalo Correia up201905348 9 hours ago</p> <p>e ser especificado numa ordem diferente de rxw (por exemplo a+wrx) é possível?</p> <p>Jose MM Cruz 9 hours ago</p> <p>Não, Pedro. Devem respeitar o ordem expressa em</p> <pre> «MODE possui o formato <u g o a><- + =><rxw>» </pre> <p>Isto é para simplificar pelas razões q escrevi acima.</p>
MODO VERBOSO	
3) No modo verboso, devemo-nos preocupar	3) Eu sei q como no xmod n se consideram

Perguntas	Respostas
<p>com permissões que não sejam `rwx` a ser mostradas na saída padrão? Por exemplo a correr com chmod (assumindo que as permissões não são alteradas) `chmod -v 0700 nome_diretorio` obtem-se na saída padrão: `mode of 'nome_diretorio' retained as 2700 (rwx--S---)` (Note-se o S) Mas se correr com a nossa implementação atual de xmod `./xmod -v 0700 nome_diretorio` obtem-se: `mode of 'nome_diretorio' retained as 0700 (rwx-----)` É isso que é pretendido com este miniprojeto? Ou a ferramenta de correção irá exigir a indicação daquele S?</p>	<p>outros tipos de "permissões", além das de "rwx", para u, g e o. Vai haver probs na comparação dos resultados do xmod com os do chmod. Por iso é q o "automatismo" na avaliação será limitado. O chmod é a vossa referência, mas há aspectos q n podem segui-la sem ir contra a especificação do xmod.</p>
<p>4) Quando corremos `chmod` em modo verboso para um ficheiro que não existe, é escrito em stdout ` ` ` chmod: cannot access 'nome_ficheiro': No such file or directory failed to change mode of 'nome_ficheiro' from 0000 (-----) to 0000 (-----) ` ` em que o valor dos modos parece ser, tanto quanto consegui detetar, undefined behavior, embora costume ser 0000 e, nos últimos dias, não tenho conseguido obter nenhum valor diferente de 0000. Qual o comportamento pretendido para `xmod`? Assumir que os modos do ficheiro que não existe é sempre 0?</p>	<p>4) Pois, estive ag a experimentar e, na verdade, nunca tinha visto este comportamento no modo "verboso". Tentem imitar a informação do chmod, como for possível - lembrem-se q tb a nossa maq de referência é o GNOMO! (na minha maq tb me deu como referiu: \$ chmod -v a+x blabla chmod: cannot access 'blabla': No such file or directory failed to change mode of 'blabla' from 0000 (-----) to 0000 (-----)</p>
<p>5) chmod faz escape de certos caracteres no nome do ficheiro, por exemplo, se corremos `chmod -v 0777 ""` a output é `mode of "" retained as 0777 (rwxrwxrwx)` é necessário replicar este escaping no `xmod`?</p>	<p>5) Como foi agora dito na Adenda (para teste e avaliação) usaremos sempre nomes "bem comportados", à excepção de um teste. Isto é para q quem n tiver tempo de imitar o comportamento do chmod a este respeito, só perderá um pouquinho de pontuação! Eu n vi bem as idiosincrasias do chmod, como a q o Pedro apresentou, mas posso lhe dizer q, no geral das aplicações em Unix, se tiverem cuidado com a maneira como processam programaticamente os nomes dos ficheiros q são indicados como parâmetros, raramente (ou nunca) terão probs em lidar com qq nome q o utilizador indique. Pedro Gonçalo Correia up201905348 9 hours ago Sim, o problema não é na execução do programa</p>

Perguntas	Respostas
	<p>para ficheiros com nome "estranho", mas sim na apresentação da saída padrão do modo verboso (onde nós estamos a escrever "" porque o nome do programa é ' , mas o chmod faz um escape estranho ficando "\"). Não encontrei em lado nenhum as regras deste escaping</p> <p>Jose MM Cruz 8 hours ago</p> <p>Pedro, isso é realmente estranho e eu nem consigo reproduzir o q mostrou (mais abaixo mostro-lhe o q experimentei). No entanto, acho q n devem tb gastar mto tempo com isto: pode estar certo de q se se verificar alguma esquisitice na saída do chmod q seja mto difícil reproduzir no xmod (até pq pode nem fazr sentido!), n haverá penalização na avaliação.</p> <p>Pedro Gonçalo Correia up201905348 8 hours ago</p> <p>está aqui um exemplo já agora, por curiosidade</p> <p>image.png</p> <p>image.png</p> <p>Jose MM Cruz 8 hours ago</p> <pre>\$ chmod -v 0777 "</pre> <p>chmod: cannot access ": No such file or directory</p> <p>failed to change mode of " from 0000 (-----) to 0000 (-----)</p> <pre>\$ chmod -v 0777 ""</pre> <p>chmod: cannot access "": No such file or directory</p> <p>failed to change mode of "" from 0000 (-----) to 0000 (-----)</p> <pre>\$ chmod -v 0777 \'</pre> <p>chmod: cannot access "': No such file or directory</p> <p>failed to change mode of "'" from 0000 (-----) to 0000 (-----)</p> <p>Mas atenção q as plicas ' ou " , se n estiverem escapadas (com \) têm de emparelhar! caso contrário a shell faz uma interpretação q n é o q pretendemos...</p> <p>Jose MM Cruz 8 hours ago</p> <p>Vi ag o instantaneo q me enviou. É estranho, pq eu n obtenho a mesmo resultado com exacta/ o mesmo nome ("")!! Mais uma vez: o GNOMo é a nossa maq de referência.</p>
PROCESSOS, REGISTO DE EVENTOS E SINAIS	
6) No enunciado diz "quando xmod é iniciado pelo utilizador, o primeiro processo a ser executado (processo	<p>6) Tenho resposta para 2 das perguntas q faz aqui; a terceira tenho q pensar no assunto...</p> <p>Jose MM Cruz 8 hours ago</p>

Perguntas	Respostas
<p>inicial) deve, antes de terminar, aguardar pela terminação de todos os restantes processos do programa;"Como todos os processos executam uma nova instância de xmod igual em tudo exceto em FILE/DIR, não conseguimos arranjar forma de distinguir o processo inicial, portanto fizemos com que todos os processos esperem pelos respetivos filhos (o que indiretamente faz com que o inicial espere por todos). É essa a solução pretendida, ou só o inicial devia esperar?Pela mesma razão, estamos a usar o `instant` do registo de eventos relativo ao início do processo em questão, e não do programa. Pode dar uma dica em como poderíamos obter o instante em que o programa (e não o processo) começou a correr?Segundo o enunciado, se o ficheiro LOG_FILENAME já existir deve ser truncado. Mas como não é possível distinguir cada instância, os processos filhos acabam por truncar o que o pai escreveu. Todos estes problemas parecem estar interligados, e leva-nos a crer que há algum truque que nos está a falhar, mas não conseguimos descobrir qual.</p>	<p>Processo inicial esperar pela terminação de todos os outros: veja pf a última linha da tabela https://web.fe.up.pt/~jmcruz/so/so.2021-i/doc/unix-freq-probs.html</p> <p>Como conseguir q um processo filho não trunque o ficheiro de registo criado pelo processo original: pensem num "truque" para q um processo abaixo na árvore saiba se é o original ou se é um descendente; ou, de outra maneira, pensem como pode o proc original informar os descendentes q não são o original! Usem os meis (legais!) à vossa disposição para fazer essa transmissão de informação.</p> <p>Sobre a questão dos tempos a registar, contados a partir do tempo de início: dexe-me pensar... Posso ter avalaido mal o requisito...</p> <p>Jose MM Cruz 8 hours ago</p> <p>O resto das respostas terá de ficar para mais tarde: ou ainda de manhã ou, de certeza, à tarde.</p> <p>Pedro Gonçalo Correia up201905348 8 hours ago</p> <p>Ok, professor, obrigado.</p> <p>Jose MM Cruz 8 hours ago</p> <p>Mas olhe q apreciei bastante o cuidado com você e o seu grupo estão a dedicar ao projecto! :smile:</p> <p>Pedro Gonçalo Correia up201905348 8 hours ago</p> <p>Quanto ao wait, é exatamente isso que estamos a usar, mas pelo que percebemos, isso significa que cada processo vai ter de esperar por pelos respetivos filhos; ou seja para além de o processo inicial esperar por todos os outros, cada processo filho vai esperar pelos filhos deste. Correto, ou estou a interpretar mal o uso do wait?</p> <p>Jose MM Cruz 8 hours ago</p> <p>Sim, Pedro. cada pai vai ter de esperar pelos seus filhos. E repare q um pai não pode esperar directamente pelos netos, bisnetos, etc...</p> <p>Jose MM Cruz 1 hour ago</p> <p>6) continuação: os processos abaixo do inicial têm de saber qdo o programa começou para poderem escrever no fich de registos tempos (1º campo de cada linha) com algum significado. Neste momento, n consigo pensar em mais de 2 possibilidades:</p> <ul style="list-style-type: none"> - o processo inicial, de alguma forma, passa aos seus descendentes a informação de qdo

Perguntas	Respostas
	<p>começou a executar (de qq maneira ele já tinha de passar aos descendentes a informação de q era o 1º, o inicial!) (eu queria evitar esta forma, mas desconfio q é mesmo a melhor - mais fácil);</p> <p>- a 2ª forma é obter do sistema operativo a info do tal tempo (em Linux):</p> <p>1º obter do 1º processo o seu pid, pid1º</p> <p>2º ler proc/pid1º/stat</p> <p>3º perceber o q se obtém da 22ª palavra mostrada</p> <p>(man proc; pesquisar starttime)</p> <p>Como vêem, esta forma é só para "os destemidos", q querem seguir o caminho mais difícil. :slightly_smiling_face:</p>
<p>7) O evento PROC_CREAT deve ser escrito pelo pai (registar que vai criar/criou um novo processo) ou pelo filho (registar que acabou de ser criado)?</p> <p>O processo inicial também deve gerar esse evento? (ou seja ser logo o primeiro registo a fazer mal o programa comece a correr)</p>	<p>7) Este aspecto poderia ter sido melhor explicado, no entanto o enunciado diz q cada linha do fich de registos é preenchida pelo processo em q ocorre o evento q o afecta. Assim, deverá haver linhas para os processos q são criados e linhas para os q criam. Numa análise posterior ao fich, tais linhas devem emparelhar, excepto a do processo inicial.</p>
<p>8) O programa deve estar preparado para registar o evento SIGNAL_RECV para qualquer sinal possível, mesmo que seja um sinal que de outro modo nunca seria usado pelo programa?</p>	<p>8. Sim, todos os sinais recebidos devem ficar registados (apesar de o único de interesse no enunciado ser o SIGINT). Note-se, contudo, q a qse totalidade dos sinais n tratados n dará tempo ao processo de escrever coisa alguma nos registos!... :wink:</p>
<p>9) Estamos a ter dificuldade a converter um signo para o seu nome (por exemplo SIGINT). Pensávamos que a função char *sigabbrev_np(int sig); trataria disso mas aparentemente ela não está disponível em todas as versões (notavelmente testei no meu computador e no GNOMO e não existia). Ou isso ou estamos a fazer alguma coisa mal...</p>	<p>9) Vejo q pesquisou mais do q eu: eu n conhecia tal função (e duvidaria q fosse POSIX). Tente strsignal(), q é POSIX1</p> <p>[Uma outra opção seria psignal() mas tem um problemazinho q complica mais a tarefa...]</p>
<p>10) O evento FILE_MODF deve ser registado se as permissões forem alteradas com sucesso mas as antigas permissões sejam iguais às novas permissões (ambos 0777 por exemplo)? Nesse mesmo cenário, `nfmod` deve ser incrementado?</p>	<p>10) Pois, o caso q apresenta é curioso (e n foi previsto). O q eu faria (e é sempre dito no enunciado! :slightly_smiling_face:) é usar o chmod como referência. De "man chmod":</p> <p>-c, --changes : like verbose but report only when a change is made</p> <p>Mas eu, descrente como sou, confirmei: fazendo chmod -c MODE FILE/DIR em q o FILE/DIR já tinha as permissões indicadas em MODE, verifiquei q a saída do comando era.....(nada!)</p> <p>:slightly_smiling_face:</p> <p>Por isso, eu nada registaria nessa situação!</p> <p>Tb n incrementaria o 'nfmod'.</p>

Perguntas	Respostas
<p>11) Temos algumas dúvidas relativamente ao que incluir na contagem de <code>`nftot`</code>. Á partida, parece-nos razoável incluir todos os ficheiros regulares e links simbólicos.</p> <p>Mas os diretórios, deverão ser incluídos no <code>`nftot`</code> do processo pai apenas (que, ao encontrá-lo, vai criar o filho para alterar as suas permissões), no processo filho apenas (que vai alterar as suas permissões, abrí-lo e alterar as permissões dos ficheiros que estão dentro), ou ambos?</p>	<p>11) Eu faria ao registo só num deles, talvez no pai. Caso contrário, o nº total de fichs "encontrados" seria diferente de uma mera contagem de fichs feita com outra ferramenta (e.g. <code>ls -R dir wc -l</code>)</p>
OUTROS	
<p>12) No caso de invocações que são válidas em <code>`chmod`</code> mas não em <code>`xmod`</code> (por exemplo, existencia de mais argumentos que o esperado), a mensagem de erro escrita em <code>stderr</code> fica ao nosso critério para escrevermos o que quisermos?</p>	<p>12) Sim, n há qq requisito a ese respeito, excepto... o q é normal fazer-se em Unix e de já vos falei na teórica: mensagens de erro ou notificações q n corrspondem aos resultados normalmente esperados devem ser colocados no... saída padrão de erros.</p> <p>Mas, pela pergunta, vejo q já sabe isso. :-)</p>