



Ciências
ULisboa

Faculdade de Ciências da Universidade de Lisboa

Programação Paralela e Concorrente

José Eduardo Madeira

fc51720

Assignment 4

Neste trabalho de Renaming files and folders com Actor Model with Supervision Trees, utilizei a library em java dada pelo professor. Para abordar este trabalho, usei três tipos de Actors, sendo eles, um Cliente, um FolderManager e FolderWorker, cada um com a sua função. Para além dos Actors usei 3 tipos de Mensagem, StartRenamingFilesRequestMessage, RenameFileOrderMessage e a ResponseMessage.

Na classe main enviamos uma mensagem ao Client com o Request (StartRenamingFilesRequestMessage) onde estaria a Directory principal (.\tree), uma String para ser substituída, e outra String para a Substituir e por sua vez o cliente esta encarregue de enviar uma mensagem (RenameFileOrderMessage) ao FolderManager onde estaria os mesmo argumento que a mensagem que o Client recebeu, mais o seu Address; o FolderManager quando recebia esta mensagem cria dinamicamente FolderWorkers, dependendo que quantos Files, a Directory recebida tinha, e “reencaminhava” a mensagem recebida para cada um dos workers criados, porem com um Directory diferente para cada um, que seria os subfiles da Directory que o Manager tinha recebido por parte do Cliente. Por sua vez a maneira de como o FolderWorker lida com a mensagem, foi um pouco mais intrigante, tentei abordar de duas maneira possíveis, uma dela foi o facto de cada FolderWorker estar encarregue do file que foi dado pela mensagem e também dos seus subfiles, caso existissem, porem nesta abordagem não estaria a tirar partido dos benefícios do Actor Model, pois poderia acontecer um FolderWorker encarregue de milhares de subfiles, então a segunda abordagem, foi a criação de um “subManager” que seria um Actor do tipo Manager do file recebido na mensagem caso ele tivesse subfiles, e desta forma se tivesse subfiles, este “subManager” iria criar mais FoldersWorkers dinamicamente para tratar desses files. No final o FolderWorker envia uma mensagem ao Client de como o trabalho já foi feito.

Para enviar a Message SystemKillMessage() para todos os atores, dou um tempo maximo sobe o qual o programa pode correr, por exemplo 10 segundo. Ao tentar lidar com o facto da não alteração do nome, caso já houvesse um File com esse mesmo nome, existe sempre um file.txt de cada subpasta do .\tree que não muda de nome, não consegui corrigir este bug com a segunda abordagem que tomei como o handleMessage do FolderWork, pois com a primeira abordagem isso não acontecia.

Exemplo deste bug trocando o "1" por "13":

tree

- folder_13
 - file_1.txt
 - file_130.txt
 - file_1300.txt
 - file_1313.txt
 - file_132.txt
 - file_133.txt
 - file_134.txt
 - file_135.txt
 - file_136.txt
 - file_137.txt
 - file_138.txt
 - file_139.txt

folder_130

- file_1.txt
- file_130.txt
- file_1300.txt
- file_1313.txt
- file_132.txt
- file_133.txt
- file_134.txt
- file_135.txt
- file_136.txt
- file_137.txt
- file_138.txt
- file_139.txt
- file_2.txt

folder_2

- file_1.txt
- file_130.txt
- file_1300.txt
- file_1313.txt
- file_132.txt
- file_133.txt
- file_134.txt
- file_135.txt
- file_136.txt
- file_137.txt
- file_138.txt
- file_139.txt
- file_2.txt