



Ciências
ULisboa

Nome da Cadeira
2021/2022
Mestrado em Engenharia Informática

Report - Assignment 4

Renaming files and folders

Tomás Dias nº57582

Para a conceção deste sistema, utilizou-se a biblioteca em *Java* lecionada durante as aulas teóricas que implementa o modelo de computação concorrente *Actor Model*. De forma resumida, este modelo centra-se na utilização de atores (uma unidade fundamental de computação) que se encontram isolados por natureza e que por isso trabalham de forma assíncrona. Cada ator possui o seu próprio estado privado e uma mailbox na qual pode receber mensagens de outros atores, permitindo a sua comunicação. Estas mensagens podem ser processadas pelo ator, uma de cada vez, de forma a alterar o seu estado.

Na biblioteca utilizada, a implementação dos atores é feita através do uso de threads.

No que diz respeito ao desenho da arquitetura do sistema, pensou-se em definir as classes de atores:

- *Customer*, responsável por encaminhar o trabalho a ser feito a um *Manager* e receber a informação se o mesmo foi finalizado.
- *Manager*, responsável por distribuir o trabalho pelos *Employees*.
- *Employees*, responsável por executar o trabalho e enviar os resultados ao *Customer*.

A escolha recaiu sobre este modelo, e mais especificamente o modelo *Actor Model* implementado nas aulas teóricas, dada a atenção que obteve durante o decorrer das mesmas, pela sua simplicidade em comparação com frameworks mais robustos como é o caso do framework *Akka* para *Java* e por não necessitar de uma curva de aprendizagem tão elevada como no caso da utilização de *co-routines*.

Em relação à paralelização, tentou-se equacionar o problema de modo a tratá-lo como uma árvore (neste caso, a diretoria root seria a raiz dessa árvore), em que cada nó filho é processado em diferentes threads (no caso, atores), e assim sucessivamente.

Cada ator com o papel de *Employee* trata cada operação de escrita (nomeadamente, de criar, renomear ou eliminar) sobre um ficheiro de forma sequencial, pois apenas existe um único employee para cada ficheiro encarregado destas operações no decorrer do programa, evitando possíveis alterações de outros atores.

Devido às características do modelo utilizado mencionadas anteriormente, o comportamento assíncrono existente, ou seja, a não necessidade que as threads têm de esperar por outras threads, permite que todo o processamento seja efetuado sem a exigência de operações síncronas.

Houve uma tentativa de encarar o contexto recursivo do problema que foi através da utilização de um parâmetro de controlo da profundidade (novamente recorrendo à avaliação do problema como uma árvore), sendo o *path* a ser executado percorrido em função deste parâmetro.

Nota: Para executar o programa corretamente, não esquecer passar como argumentos a diretoria, o carácter a ser substituído e o carácter que o vai substituir.