



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Gestão de Stocks
Grupo N.º 2

José Ferreira (A83683)

Emanuel Rodrigues (A84776)

João Teixeira (A85504)

11 de Maio de 2019

Conteúdo

1	Introdução	3
2	Módulos	4
2.1	Manutenção de Artigos	4
2.2	Servidor de Vendas	4
2.3	Cliente de Vendas	4
2.4	Agregador	4
3	Arquitetura e Solução do Projeto	5
3.1	Manutenção de Vendas	5
3.2	Servidor de Vendas	5
3.3	Cliente de Vendas	5
3.4	Agregador	6
4	Cache de Precos	7
5	Agregador Concorrente	8
6	Sincronizacao Artigos Stock	9
7	Conclusão	10

Capítulo 1

Introdução

O objetivo deste projeto é construir um sistema de gestão de stocks, de forma a ser capaz de armazenar informação de artigos, stocks e vendas, aplicando uma arquitetura *Cliente, Servidor* com recurso a *System Calls* e vários processos, aplicando assim os conhecimentos sobre os Sistema Operativos baseados em Unix adquiridos nas aulas desta UC. Como objetivo, é necessário implementar um sistema que permita a utilização concorrente de um servidor por vários clientes, assegurando a conservação de toda a informação introduzida no sistema, bem como a consistência desta entre os vários ficheiros e cache.

Ao longo deste relatório vamos descrever as nossas abordagens a estes problemas.

Capítulo 2

Módulos

2.1 Manutenção de Artigos

O modulo da Manutenção de artigos tem como função ser um ‘preçário’ tendo informação dos diferentes produtos existentes, bem como o seu preço. Aqui é possível a criação de novos artigos, bem como a sua modificação. Sendo um modulo referente a manutenção, também é possível correr o agregador a pedido.

2.2 Servidor de Vendas

O Servidor de vendas trata todos os pedidos dos Clientes de vendas, quer sejam vendas ou entradas de stock, ou informações sobre o preço e stocks existentes no momento.

2.3 Cliente de Vendas

O Cliente de Vendas é a interface que o utilizador final usa para realizar todas as vendas e pedidos de informação, comunicando com o servidor de vendas.

2.4 Agregador

O Agregador tem como finalidade agregar vendas, ou seja, juntar todas as vendas referentes ao mesmo artigo.

Capítulo 3

Arquitetura e Solução do Projeto

3.1 Manutenção de Vendas

A Manutenção de Vendas trata de manter artigos, respetivos preços e nomes. Para manter essa informação usa dois ficheiros, *strings* e *artigos*. O utilizador consegue adicionar e modificar artigos através do *stdin*. Embora a manutenção de vendas seja independente dos restantes módulos, este comunica com o Servidor de Vendas, caso este esteja a correr, de forma a manter a sincronização da informação sobre o preço e numero de artigos existentes em ambos os lados.

Aqui também é possível correr o agregador sobre o ficheiro de vendas.

3.2 Servidor de Vendas

O Servidor de Vendas trata dos pedidos de todos os clientes, e trata de manter informação sobre o stock de cada artigo. Para manter essa informação utiliza o ficheiro *stocks*. O Servidor corre em background, consistindo em três diferentes processos:

- Um que trata da sincronização com a parte da Manutenção de Artigos (preços e criação de novos artigos), com a finalidade de manter, tanto o ficheiro de stocks, como a cache de preços sempre atualizada.
- Um que contém a cache de preços.
- Um responsável pela comunicação e tratamento dos pedidos dos clientes.

Toda a comunicação com o servidor é feita com recurso a *Named pipes*, existindo um pipe para os clientes efetuarem pedidos e um para o Manutenção de Artigos enviar as alterações que ocorram da parte dele.

3.3 Cliente de Vendas

O Cliente de Vendas interage com o utilizador a partir do *stdin* e *stdout*, e interage com o Servidor de Vendas enviando informação para o pipe de entrada que o Servidor cria, e recebe a resposta deste através de um pipe que ele cria, tendo como identificador o seu *PID*, de forma a garantir que as informações não são lidas por outro cliente. Para isto acontecer, para além do pedido, é escrito no pipe de leitura do servidor o identificador do cliente, que este está preparado para conseguir responder no pipe adequado.

3.4 Agregador

O agregador recebe input do *stdin* linhas com o formato estipulado para o ficheiro *vendas*, calculando os totais de cada artigo, devolvendo esses totais no mesmo formato pelo *stdout*.

Capítulo 4

Cache de Precos

A cache de preços, parte integrante do Servidor de Vendas, existe para evitar consultar o ficheiro de Artigos a cada pedido do cliente. Para a manter sempre sincronizada com a informação que existe na parte da Manutenção de artigos, optamos por implementar esta num processo distinto do Servidor em si, para não causar atrasos a resposta dos pedidos de clientes. Assim, quando o servidor necessita da informação sobre um preço de um artigo, envia o pedido através do pipe criado para o efeito.

Capítulo 5

Agregador Concorrente

O Agregador Concorrente, implementado na Manutencao de Artigos, consiste em correr o agregador sobre o ficheiro vendas, dividindo-o em partes, sendo capaz de agregar essas pequenas partes em paralelo e por fim agregar as partes mais pequenas todas, tornando assim a agregacao deste ficheiro muito mais rapida.

Capítulo 6

Sincronizacao Artigos Stock

Como a informacao contida no ficheiro Stocks tem que estar sempre atualizada com o ficheiro Artigos, ao ser criado o ficheiro Artigos, e guardada a informacao de quando foi criado. Esta mesma marca temporaria e transcrita para o ficheiro Stocks na sua criacao. Para assegurar que o ficheiro Stocks tem a informacao referente ao ficheiro Artigos mais atualizado, caso essa marca difira entre ficheiros, o ficheiro Stocks e totalmente refeito. Tambem quando sao adicionados novos artigos, essas alteracoes sao comunicadas ao Servidor, que trata de atualizar o ficheiro. Quando o servidor nao esta a correr, essa verificacao e feita ao arranque do mesmo, tendo assim a certeza que as referencias a artigos do ficheiro Stocks sao corretas e as mais recentes.

Capítulo 7

Conclusão