

Gestão de Vendas

Relatório do trabalho prático

Grupo 98

Joana Cruz(A76270)
Etienne Costa (A76089)
Rui Azevedo (A80789)

Maio 2019



Universidade do Minho
Escola de Engenharia

Sistemas Operativos
Mestrado Integrado em Engenharia Informática
Universidade do Minho

1 Introdução

Este projeto surge no âmbito da Unidade Curricular de Sistemas Operativos e tem como objetivo a implementação de um sistema de gestão de inventário e vendas.

O sistema é constituído por vários programas: manutenção de artigos, servidor de vendas, cliente de vendas e agregador de dados, todos estes programados na linguagem de programação C.

É pretendido com este trabalho explorar os diferentes mecanismos de um sistema operativo *UNIX*.

2 Proposta de Arquitetura

O sistema desenvolvido conta com quatro entidades principais, manutenção de artigos, servidor de clientes, clientes de vendas e agregador de vendas. Todas estas entidades comunicam entre elas de modo a criar um sistema de gestão de vendas.

O sistema desenvolvido tem como base um modelo de interacção por diálogo, isto é, quando um cliente se pretende ligar ao servidor, é mandado um pedido de conexão ao mesmo. Entretanto o programa cliente cria dois *fifos* um para enviar pedidos e outro para responder. Quando o servidor recebe o pedido de conexão, cria um processo filho de maneira a atender o novo cliente. Esse processo liga-se aos *fifos* criados pelo cliente e envia uma mensagem de sucesso que representa o início de comunicação entre cliente-servidor.

Todos os pedidos de consulta a ficheiros circulam pelos *fifos* reservados para os clientes sendo que podem existir vários clientes a fazer várias consultas em simultâneo.

Todos os pedidos de implicam alteração de ficheiros são enviados pelo *fifo* público criado servidor para serem tratados sequencialmente, fazendo com que não haja conflitos de escrita de dois processo diferentes no mesmo processo. Estes tipos de pedidos correspondem a vendas, actualizações de stock e pedidos de agregação.

3 Manutenção de artigos

Nesta implementação, só é possível haver uma instância do programa de manutenção de artigos.

Este programa tem como objectivo responder a quatro comandos:

- i nome preço : inserir artigos dado um nome e um preço.
- n cod nome : actualizar um nome de um determinado artigo.
- p cod preço : actualizar o preço de um determinado artigo.
- s cod : dispor a informação de um determinado artigo.
- a : manda pedido ao servidor para executar a agregação.

3.1 Inserir artigos

Os novos artigos são inseridos sempre no fim do ficheiro *"artigos"* assim como o nome do artigo é inserido sempre no fim do ficheiro *"strings"*. De modo a guardar a referência do nome do artigo, foi usado o valor de retorno da função *lseek*. Antes de inserir o nome do artigo, reposicionámos o descritor de ficheiro no fim do ficheiro *"strings"*. É neste ponto, antes de inserir o nome do artigo, que é guardada a referência do nome e, posteriormente, este mesmo escrito no ficheiro. Só após esta acção é que o artigo é guardado no ficheiro *"artigos"*.

3.2 Actualização de nomes de artigos

O método utilizado para actualizar nomes é muito semelhante ao de inserir artigos.

O novo nome é inserido no fim do ficheiro *"strings"* e, anteriormente, é guardada a nova posição onde foi inserido, uma vez mais, através do valor de retorno da função *lseek*. De seguida, o campo correspondente à referência do nome do artigo é actualizado no ficheiro *"artigos"*.

3.3 Actualização do preço de um artigo

A actualização do preço é uma tarefa mais básica sendo que só é necessário localizar o artigo pretendido no ficheiro *"artigos"* e actualizar o campo referente ao preço.

4 Cliente de vendas

O cliente de vendas permite as seguintes interacções com o utilizador:

- cod : dispõe a informação do stock e preço de um artigo
- cod quantidade : caso a quantidade seja negativa, é realizada uma venda e actualizado o stock, caso contrário apenas é actualizado o stock.

Quando um cliente é criado são criados dois *fifos*, um para enviar pedidos ao servidor e outro para receber respostas aos pedidos. Nestes *fifos* apenas são enviados pedidos de consulta aos diferentes ficheiros, fazendo assim um canal de comunicação reservado para cada cliente. O resto dos pedidos, que implicam escrita nos ficheiros, são enviados pelo *fifo* público criado pelo servidor.

4.1 Dispor a informação do stock e preço

Este tipo de comandos são enviados através dos *fifos* que cada cliente dispõe para si. Desta maneira, conseguimos ter vários clientes a fazer várias consultas a artigos simultaneamente.

Uma vez que este tipo de comando só permite fazer consultas ao ficheiro "*artigos*", não existe problema em ter vários clientes a fazer estas consultas simultaneamente.

4.2 Realizar vendas e actualizações de stocks

Uma vez que este comando implica alterar informação em disco dos diferentes ficheiros, este tipo de pedidos são enviados pelo *pipe* público de modo a que o servidor de vendas trate esta informação sequencialmente não comprometendo a veracidade dos ficheiros.

O servidor de vendas processa a informação pedida e envia resposta pelo *pipe* de resposta do cliente.

5 Servidor de vendas

O servidor de vendas vai estar constantemente à escuta do *fifo* público à espera de pedidos para processar informação.

Caso receba um pedido de conexão de um cliente, é criado um processo filho, que, por sua vez, liga-se aos *fifos* criados pelo cliente. Desta maneira, o servidor dispõe de um "funcionário" para cada cliente.

Para além disso, o servidor pode também receber pedidos para executar vendas e/ou actualizações de stocks. Estes pedidos são executados sequencialmente pelo servidor, de modo a não haver conflito de escritas no mesmo ficheiro por processos diferentes simultaneamente.

Por fim, o servidor também poderá correr o agregador a pedido da manutenção de artigos. O servidor consulta o ficheiro "*logs*" de modo a encontrar a última agregação feita e, faz a agregação de vendas a partir desse ponto, guardando o resultado num ficheiro criado com a data do pedido.

6 Agregador

O objectivo do agregador é agregar um conjunto de vendas a partir da última agregação feita.

O programa recebe pelo *stdin* o conjunto de dados a agregar e, a partir daí, vai lendo venda a venda e adiciona a um ficheiro temporário a agregação. Isto é feito através da função *lseek*. É lida a venda a agregar, executa-se a função `lseek(fd,codigo * sizeof(struct venda), SEEKSET)` e, caso já exista informação nessa espaço, é lida e somada a quantidade e montante à venda a adicionar, caso ainda não exista informação, é inserida directamente a venda no ficheiro.

Ao fim de percorrer as vendas a agregar, é percorrido o ficheiro temporário criado para a agregação limpando os espaços em branco, e, por fim, essa informação é enviada para o *stdout*.

7 Conclusões