

Universidade do Minho

GESTÃO DE VENDAS

MESTRADO INTEGRADO EM
ENGENHARIA INFORMÁTICA

SISTEMAS OPERATIVOS

(2º ANO, 2º SEMESTRE, 2018/2019)

A86435 Miguel André da Silva Solino

Braga
Maio 2019

Índice

1	Introdução	2
2	Descrição do Problema	3
2.1	Manutenção de Artigos	3
2.1.1	Inserir um artigo	3
2.1.2	Alterar nome de um artigo	3
2.1.3	Alterar preço de um artigo	3
2.1.4	Agregar as vendas	3
2.2	Servidor de Vendas	4
2.3	Cliente de Vendas	4
2.3.1	Vizualizar stock de um artigo	4
2.3.2	Alterar nome de um artigo	4
2.4	Agregador	4
3	Estruturas	5
3.1	Artigos	5
3.2	Stock	5
3.3	PFifo	5
3.4	Info	6
4	Execução de comandos	7
4.1	Processo Servidor de Vendas (sv)	7
4.2	Processo Manutenção de Artigos (ma)	7
4.3	Processo Cliente de vendas (cv)	7
4.4	Processo Agregador (ag)	7
5	Conclusões e Trabalho Futuro	8

1. Introdução

Este trabalho foi realizado no âmbito da unidade curricular sistemas operativos e tem como objetivo o processamento de ficheiros.

Este processamento foi implementando com recurso a *system calls* e à criação de alguns processos para que fosse possível o paralelismo.

2. Descrição do Problema

Os 4 programas que foram realizadas são:

- Manutenção de Artigos
- Servidor de Vendas
- Cliente de Vendas
- Agregador

Estes são programas que correm individualmente exceto o Servidor de Vendas que corre em segundo plano e recebe informações/comandos dos outros 3 para poder agir.

2.1 Manutenção de Artigos

Programa que permite inserir novos artigos e depois alterá-los

2.1.1 Inserir um artigo

Se, o input for um 'i' seguido de uma string e um número, então o programa irá adicionar um artigo novo com a string como nome, o número como preço e colocará no STDOUT o código do artigo criado.

2.1.2 Alterar nome de um artigo

Se, o input for um 'n' seguido de um número e uma string, então o programa irá alterar o nome do artigo com o número como código e a string como nome.

2.1.3 Alterar preço de um artigo

Se, o input for um 'p' seguido de um número e outro número, então o programa irá alterar o preço do artigo com o número como código e o segundo número como preço.

2.1.4 Agregar as vendas

Se, o input for um 'a' o programa chama o programa agregador e agrega o ficheiro das vendas que foram feitas até agora.

2.2 Servidor de Vendas

Programa que recebe comandos e pedidos dos outros e age conforme o que for pedido.

2.3 Cliente de Vendas

Programa que permite fazer vendas dos artigos e mostrar stocks dos mesmos.

2.3.1 Vizualizar stock de um artigo

Se, o input for um número então o programma assume esse número como um código e contata o servidor de vendas para lhe fornecer o stock e preço do artigo com esse código.

2.3.2 Alterar nome de um artigo

Se, o input for um número seguido de outro número, o programma assume esse comando como uma venda e diz ao servidor que uma venda foi feita com o primeiro número como código e o segundo número como a quantidade vendida.

2.4 Agregador

Programa permite agregar vendas e mostra a agregação das mesmas no final no STDOUT.

3. Estruturas

3.1 Artigos

Os artigos são organizados e guardados em uma estrutura que guarda todos os campos do mesmo.

A estrutura em si é um inteiro e um float que correspondem à posição do nome do artigo no ficheiro strings e o preço, respetivamente.

```
struct Artigo{  
    int posicao;  
    float preco;  
};
```

Figura 3.1: Representação em memória de um artigo

3.2 Stock

Os stocks são organizados e guardados em uma estrutura que guarda todos os campos do mesmo.

A estrutura em si é um inteiro que corresponde ao código do artigo e outro inteiro que corresponde ao stock do mesmo.

```
struct Stock{  
    int codigo;  
    int stock;  
};
```

Figura 3.2: Representação em memória de um stock

3.3 PFifo

Quase toda a informação que é enviada para e recebida do servidor de vendas nos pedidos é enviada através desta estrutura.

A estrutura em si é um número de processo do programa que realizou o pedido, um inteiro que corresponde ao código do artigo, um inteiro que corresponde à quantidade da venda realizada, um char que corresponde ao comando que foi pedido no programa e o um float que corresponde ao preço do artigo.

```
struct PFifo{  
    pid_t fifoname;  
    int cod;  
    int quantidade;  
    char comando;  
    float preco;  
};
```

Figura 3.3: Representação em memória de um PFifo

3.4 Info

Toda a informação que é devolvida do servidor de vendas para o cliente de vendas vem numa estrutura deste tipo.

A estrutura em si é um int que corresponde ao novo stock do artigo e um float que corresponde ao preço do artigo em questão.

```
struct Info{  
    int stock;  
    float preco;  
};
```

Figura 3.4: Representação em memória de um Info

4. Execução de comandos

4.1 Processo Servidor de Vendas (sv)

O servidor de vendas é sempre o programa a ser executado pois é o que fica em segundo plano. No momento em que se inicia ele fica parado à espera de receber um pedido dos outros programas. Após o pedido ser feito ele analisa qual foi o comando enviado e processa a informação recebida conforme o mesmo. No momento de enviar, caso para quem vá enviar seja o cliente de vendas, então envia por um pipe único criado pelo cliente para que seja possível serem corridos ao mesmo tempo vários clientes.

4.2 Processo Manutenção de Artigos (ma)

A manutenção de artigos logo após ser executada fica à espera de input. Dependendo desse input o programa envia um pedido para o servidor ou não. No caso em que seja para adicionar um artigo, então avisa o servidor que foi criado um artigo novo e pede ao mesmo que inicialize o stock do mesmo. Nos outros três comandos não contata o servidor sendo independente de si, exceto no último, que mesmo em parte sendo independente, executa o agregador.

4.3 Processo Cliente de vendas (cv)

O cliente de vendas é parecido com a manutenção de artigos mas este está sempre em contato com o servidor. Como este programa pode ser corrido simultaneamente com outro igual então ele cria um pipe único para cada cliente. Assim, quando for a enviar e a receber informação do servidor, nenhuma informação se perde nem é entregue por engano.

4.4 Processo Agregador (ag)

O agregador é um programa totalmente independente que faz parse de vendas. Recebe vendas pelo STDIN e devolve para o STDOUT o resumo de tudo sem qualquer execução de comandos. A única coisa que recebe são vendas introduzidas pelo utilizador ou redirecionando um ficheiro utilizando a bash.

5. Conclusões e Trabalho Futuro

Em suma, considero que o trabalho foi realizado na sua totalidade, de forma estruturada, respondendo a todos os requisitos, exceto os aspectos valorizados que estavam presentes no enunciado.

Um aspeto que poderia ser melhorado era adicionar datas às vendas e permitir a escolha de agregação entre datas.