

Universidade do Minho
Departamento de Informática
Mestrado Integrado em Engenharia Informática

Sistemas Operativos

2018/2019

Gestão de Venda

Grupo 20



Gonçalo Pinto
83732



José Pinto
84590



Luís Lopes
85367

Índice

1. Introdução
 - 1.1 Objetivos Gerais
2. Análise e Especificação do Problema
 - 2.1 Descrição formal do problema
3. Desenho e Conceção
 - 3.1 Arquitetura do Sistema
 - 3.2 Manutenção de Artigos
 - 3.3 Cliente de Vendas
 - 3.4 Servidor de Vendas
 - 3.5 Agregador
4. Implementação
 - 4.1 Comunicação Cliente Vendas, Servidor Vendas
 - 4.2 Comunicação Manutenção de Artigos, Agregador
 - 4.3 Funcionamento da Manutenção de Artigos
 - 4.4 Funcionamento do Cliente Vendas
 - 4.5 Funcionamento do Servidor Clientes
 - 4.6 Funcionamento do Agregador
 - 4.7 Makefile
5. Conclusão

Capítulo 1

Introdução

No 2º ano do Mestrado Integrado em Engenharia Informática, no ano letivo 2018/2019, a unidade curricular *Sistemas Operativos* apresenta como proposta, na componente prática, a realização de um programa para Gestão De Vendas. O Sistema terá como finalidade a manutenção de artigos, servidor de vendas, cliente de vendas, e agregador de dados.

1.1 Objetivos Gerais

Este projeto tem como objetivo principal consolidar e colocar em prática os conceitos fundamentais transmitidos durante a UC de Sistemas Operativos. Com o presente trabalho pretendesse desenvolver uma aplicação com recurso a processos concorrentes, controlando concorrentemente a iteração entre eles através de pipes, forks, execs entre outras técnicas.

Capítulo 2

Análise e Especificação do Problema

2.1 Descrição formal do problema

Como referido no capítulo anterior, é pretendido realizar um programa de Gestão de Vendas. O programa vai ser constituído por 4 aplicações, que serão executadas no terminal:

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

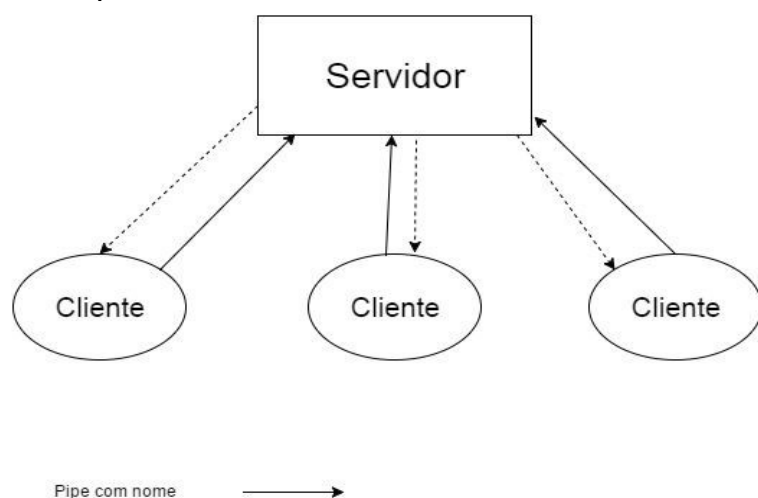
Capítulo 3

Desenho e Conceção

3.1 Arquitetura do Sistema

O sistema desenvolvido é baseado numa arquitetura cliente/servidor. Desta forma todas as operações que o cliente pretenda efetuar são executadas através do servidor.

De seguida apresenta-se uma imagem que pretende representar a comunicação do cliente para o servidor e as suas respostas.



3.2 Manutenção de Artigos

A manutenção de artigos, construída dirigida aos administradores do sistema, consiste num sistema com uma série de comandos que vai permitir ao administrador adicionar produtos ao sistema, alterar o nome e quantidade de produtos já existentes e permite também agregar as vendas que se encontram no ficheiro VENDAS.txt. Quando se adiciona um produto ao sistema, este recebe um código gerado pelo programa que é guardado no ficheiro ARTIGOS, onde consta além do código, o preço do artigo. Entretanto o nome do artigo é guardado por sua vez num ficheiro denominado STRINGS. Alguns comandos que se podem executar no Manutenção de Artigos:

i <nome> <preço>

n <código> <novo produto>

p <código> <novo preço>

a

3.3 Cliente de Vendas

O Cliente de Vendas permite ao utilizador adicionar vendas efetuadas quando a quantidade é negativa, ou reposições de Stock quando a quantidade introduzida for positiva e mostrar a quantidade em stock e preço de um dado produto. Este sistema interage diretamente com o Servidor de Vendas, fazendo executar, a pedido do utilizador, diversas funcionalidades para que o ficheiro STOCKS e VENDAS sejam atualizados. Alguns comandos que se podem executar no Servidor Vendas:

<código numérico>

<código numérico> <quantidade>

3.4 Servidor de Vendas

É constituído por várias funções que permitem atualizar o stock, encontrar stock ou preços bem como adicionar vendas ou providenciar ao agregador o ficheiro VENDAS para agregar. O Servidor de Vendas recebe ordens, tanto do Cliente de Vendas como da Manutenção de Artigos, caso seja pedido para agregar. Como este é necessário para todas as funções do Cliente de Vendas, e algumas da Manutenção de Artigos, tem de estar sempre ativo e pronto a receber ordens.

3.5 Agregador

O Agregador recebe através do Terminal uma série de vendas que por sua vez agrega imprimindo o resultado para o terminal. Este também é capaz de agregar as vendas que se encontram no ficheiro VENDAS.txt a pedido da Manutenção de Artigos. Sendo que o output é impresso não para o terminal, mas sim para um ficheiro novo, cujo nome representa a data da agregação.

Capítulo 4

Implementação

4.1 Comunicação Cliente Vendas, Servidor Vendas

A comunicação do cliente para o servidor é feita entre processos por *pipes* com nome. Onde o servidor vai criar dois *pipes* para cliente de vendas utilizar um para escrever o que pretende visualizar para posteriormente o servidor leia deste *pipe* o pedido e de seguida processar a informação após efetuada o pedido do cliente o servidor escreve no outro *pipe* que criou para o cliente de vendas ler e posteriormente ser apresentada ao utilizador a informação recebida.

4.2 Comunicação Manutenção de Artigos, Agregador

A comunicação entre estes dois só acontece quando é preciso fazer agregação do ficheiro VENDAS.txt e para tal é também chamado o Servidor Vendas que criou outros dois pipes com nome ao ser iniciado. A comunicação destes dois programas é feita entre processos por pipes com nome onde posteriormente o servidor vai fazer executar o agregador.

A Manutenção de Artigos recebe o comando de através do terminal e comunica a necessidade de agregação ao Servidor Vendas através do pipe com nome que pode escrever. O Servidor vai receber essa informação através desse mesmo pipe aí vai ler a primeira linha do ficheiro VENDAS.txt onde encontra a última linha agregada posteriormente posiciona-se no ficheiro de vendas e começa a traduzir para um ficheiro temporário pois estas estão em binário e o agregador não consegue processar em binário após esta operação o servidor vai redirecionar o input para o ficheiro criado após a tradução e o output duplicando os descritores de ficheiros e a sua aplicação no redireccionamento, mais especificamente dos descritores standard de entrada e saída para um ficheiro com a hora e dia atual de seguida é feito um fork ou seja é criado um processo filho que trata de executar o agregador através de um `execv`. O processo pai vai

esperar por este filho terminar para depois atualizar a primeira linha do ficheiro VENDAS.txt com agora a última linha agregada este número é também retornado através do outro pipe com nome criado anteriormente para ser apresentado na manutenção de artigos quantas linhas foram agregadas.

4.3 Funcionamento da Manutenção de Artigos

A manutenção de artigos está em ciclo sempre a ler do terminal tudo o que o utilizador insira pelo terminal, informação esta que tem que coincidir com os códigos acima referidos. Este programa tem uma sequência de testes para encontrar se a primeira letra que o utilizador inseriu é válido e vai de encontro aos códigos possíveis após encontrar uma correspondência é criado um processo onde o processo filho vai executar a operação que o utilizador pretende entre elas atualizar uma dada linha no ficheiro STRINGS ou ARTIGOS ou simplesmente introduzir uma entrada em cada ficheiro onde aqui é retornado o identificador do produto que inseriu. Neste programa como foi referido se o utilizador introduzir a operação que pretende agregar vendas é feita a comunicação com servidor como está acima descrito.

4.4 Funcionamento do Cliente Vendas

O cliente de vendas está em ciclo sempre a ler do terminal tudo o que o utilizador insira pelo terminal posteriormente é comunicado ao servidor de vendas como foi descrito acima. Posteriormente o programa decompõem o pedido vendo o tamanho deste para saber quantos números tem de ler do *pipe* com nome que abriu apenas para ler e apresentar ao utilizador a informação.

4.5 Funcionamento do Servidor Clientes

Cria no total 4 pipes com nome, 2 para comunicar com o programa cliente de vendas e outros 2 para comunicar com o programa de manutenção de artigos. Este tem várias funções que lhe permitem navegar pelo ficheiro STOCKS.txt e ARTIGOS.txt. Nestes ficheiros não existem espaços vazios, é tudo escrito continuamente. Então para navegarmos nestes ficheiros construímos várias estruturas que têm tamanhos fixos e permitem, portanto, usar a função lseek

essencial para operação sobre ficheiros neste caso posicionar na posição correta do ficheiro. Quando é pedido para agregar o servidor é essencial para comunicar entre o programa que mantém os artigos e o que agrega pois têm que executar o que foi acima descrito no ponto 4.2 na comunicação destes dois programas utilizando diversas técnicas lecionadas nesta cadeira.

4.6 Funcionamento do Agregador

O agregador como foi acima descrito vai receber pelo terminal linhas no formato de vendas, ou seja, um código, a quantidade e o total faturado. Este vai estar sempre a ler do terminal até que o utilizador introduza “exit” onde são apresentadas as linhas que foram introduzidas agregadas ou seja as linhas com o mesmo código é somado a quantidade como o total faturado. Para isso utilizamos um array dinâmico de estruturas de vendas onde vamos armazenando os valores. Após ler uma linha decompomos a linha criando uma estrutura de venda e preenchendo os parâmetros desta com os valores recebidos. Após isso procura-se no array se existe algum código já introduzido igual ao que foi introduzido se encontrar é somado a quantidade e total faturado. Senão encontrar é introduzido no array se este não tiver mais capacidade do que atual é realocado a memória, isto é, é criado uma array com o dobro do tamanho do anterior onde o array utilizado anteriormente é copiado para este. Por fim é apresentado a informação no array do fim do programa como também a libertação da memória utilizada pelo array.

4.7 Makefile

De seguida apresenta-se a makefile utilizada.

```
CC = gcc
CFLAGS = -Wall
all: ag cv ma sv
agregador:
    $(CC) $(CFLAGS) ag.c -o ag
cliente:
    $(CC) $(CFLAGS) cv.c -o cv
manutencao:
    $(CC) $(CFLAGS) ma.c -o ma
servidor:
    $(CC) $(CFLAGS) sv.c -o sv
clean:
    rm -f ag cv ma sv
```


Capítulo 5

Conclusão

Neste trabalho foi realizado um programa para Gestão De Vendas. A aplicação foi idealizada e desenvolvida utilizando a linguagem de programação C. Algumas melhorias poderiam ter sido feitas como *caching* de preços que consiste em guardar os artigos mais populares em memória, agregação concorrente e a compactação do ficheiro STRINGS.

Contudo o pedido do programa de manutenção para agregar não estar a funcionar por motivos que não conseguimos este programa não consegue abrir o pipe com nome designado para mandar o sinal de pedido de agregação este objetivo ficou comprometido parcialmente. Pois em testes realizados percebemos que o servidor conseguia redirecionar os descritores como foi descrito e executar o agregador contudo a comunicação da manutenção com o servidor não se efetua. Ao realizar várias experiências deparamos que o cliente não consegue processar pedidos de consulta, atualização de stock e registo de vendas tudo de uma vez mas introduzido estas operações em separado o programa consegue interagir com servidor sem problema nenhum.

Embora as dificuldades, de um modo geral os principais requisitos foram atingidos. Tal situação foi confirmada após alguns testes do programa.