

# **Sistemas Operativos**

## **”Gestão de Vendas”**

Solange Silva  
(a68694)

Ricardo Pereira  
(a68710)

Hugo Araújo

May 10, 2019

# Contents

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Decisões Tomadas</b>	<b>3</b>
<b>3</b>	<b>Ficheiros</b>	<b>4</b>
3.1	Artigo.c . . . . .	4
3.2	Ma.c . . . . .	4
3.3	Stock.c . . . . .	4
3.4	Cliente.c . . . . .	4
3.5	Servidor.c . . . . .	4
<b>4</b>	<b>Dificuldades Encontradas</b>	<b>5</b>
<b>5</b>	<b>Conclusão</b>	<b>6</b>

# Chapter 1

## Introdução

Este trabalho tem como objetivo construir um protótipo de um sistema de gestão de inventário e vendas. O sistema deverá ser construído por vários programas: manutenção de artigos, servidor de vendas, cliente de vendas e agregador de dados.

## Chapter 2

# Decisões Tomadas

Para a realização do trabalho foi decidido criar 5 ficheiros em C separadamente, um cliente, um servidor, um manutenção de artigos(ma), um que contém as funções para criação/modificação de artigos(artigo.c) e outro para a inserção/mostra de stock (stock.c). Decidimos fazer as funções de inserção/modificação dos artigos e também inserção/mostra dos stocks em ficheiros separados do servidor e do ma porque era mais acessível a interpretação do comando que é enviado. Os vários clientes vão comunicar com o servidor no FIFO publico e o servidor irá responder a cada cliente através de um FIFO que é privado ao cliente. Decidimos também quando o gestor de artigos é inicializado, alem de criar os ficheiros de ARTIGOS e STRINGS, cria tambem o ficheiro STOCK, porque depois ao inserir um artigo é escrito no ficheiro de STOCKS o seu código e 0 de quantidade.

## Chapter 3

# Ficheiros

### 3.1 Artigo.c

Neste ficheiro fizemos as funções de inserir artigo, tal como a sua alteração de nome e preço. Para gerar o código, e de acordo com o enunciado, o tamanho da entrada no ficheiro de ARTIGO é fixo. O código é gerado colocando o apontador para o fim do ficheiro. A operação que coloca o apontador no fim do ficheiro devolve o numero de bytes lidos do ficheiro dividindo este pelo tamanho fixo da entrada, obtendo assim um código único. Para fazer a pesquisa pelo código, o que fazemos é usando a operação lseek com o offset da multiplicação entre o tamanho da entrada e o código, sendo que este coloca-nos na posição pretendida.

### 3.2 Ma.c

Ao ser executado ele lê caracter a caracter do input, e quando o caracter for new line ele cria um processo filho, e esse mesmo é responsável pela execução do comando, enquanto que o processo pai fica a espera do output do comando que o filho executou. No ma, para comunicar entre o processo pai e o processo filho usamos um pipe anónimo. Quando está no processo filho, o file descriptor de leitura do pipe é fechado e fazemos um dup2 do standart output para file descriptor de escrita do pipe para que depois ao executar o comando o pai consiga através do descritor de leitura, ler o output do comando que o processo filho executou

### 3.3 Stock.c

O stock.c tem as funções de mostrar e modificar o stock consoante descrito no enunciado. Em termos de pesquisa, usa o mesmo metodo usado no ficheiro do artigo.c.

### 3.4 Cliente.c

Quando o cliente é inicializado ele cria um FIFO privado para a comunicação com o servidor para que o servidor mande a resposta do comando. Este FIFO privado tem que ser único, para que cada cliente receba o output de cada comando que mandou. Para garantir a unicidade, usamos o pid do processo no nome do FIFO. O cliente abre o FIFO publico em modo escrita para mandar os comando para o servidor e também abre o FIFO privado em modo de leitura para receber o output de cada comando.

### 3.5 Servidor.c

Quando o servidor é inicializado ele abre o FIFO publico em modo escrita/leitura para não estar sempre a fechar/abrir o FIFO publico a cada pedido do cliente.

## Chapter 4

# Dificuldades Encontradas

Apareceram várias dificuldades com o decorrer da execução do trabalho, nomeadamente no servidor, pois este estava a deixar processos defuntos quando executava os comandos, mas conseguimos resolver esse problema ignorando o sinal SIGCHLD, correndo "signal(SIGCHLD, SIG\_IGN)". Chamando isto, o SIGCHLD é ignorado pelo sistema e a entrada do processo filho é eliminada da tabela de processos. Tivemos também algumas dificuldades a ler de um ficheiro, e escrever noutra. Não conseguimos fazer um requisito do enunciado, que é o agregador, pois não fomos capazes de organizar o nosso tempo da forma mais correcta.

## Chapter 5

# Conclusão

Como resultado final obtivemos 5 programas que quando compilados, e usando 2 terminais, um com o cliente e outro com o servidor, podemos, usando o teclado no terminal (ou chamado através de um ficheiro txt já com as instruções) com o cliente a ser executado, mostrar o stock disponível ou modifica-lo. Também , executando o programa ma.c, podemos, usando o terminal ou um ficheiro txt já com as instruções pretendidas, inserir artigos, assim como alterar o seu preço e o seu nome. Quando um artigo novo é inserido, este é inserido inicialmente no STOCK com quantidade 0.