

Sistemas Operativos

2018/2019

Relatório do Trabalho Prático

Gestão de vendas

Filipe Barbosa A77252;

Hugo Ferreira A78555;

Rafaela de Pinho A77293;

Índice

Introdução	3
Descrição	4
Manutenção de artigos	4
Servidor.....	4
Cliente.....	5
Agregador	5
Alguns aspetos valorizados	5
Conclusão.....	6

Introdução

Foi proposto, no âmbito da unidade curricular Sistemas Operativos, a construção de um protótipo de um sistema de gestão de inventário e vendas.

Este relatório visa fundamentar os passos e as soluções que o grupo encontrou ao longo da concretização deste projeto.

Começamos por dividir o “problema” em três partes. As duas partes principais deste trabalho consistiram na manutenção de artigos e no cliente-servidor. A terceira parte incidiu na criação do agregador e realização dos aspetos valorizados.

Com este trabalho esperamos aprofundar os conhecimentos adquiridos nas aulas.

Descrição

Manutenção de artigos

A parte da manutenção de artigos foi a primeira a ser feita neste trabalho. Nesta fase, tivemos que criar um programa que lesse do *input* e escrevesse ou modificasse o ficheiro “artigos”, e que também escrevesse no ficheiro “strings”.

Para isso criamos as seguintes funções:

- **void trataB(char *buf, char **arg1, char **arg2, char **arg3)** : recebe o *buffer* que lê do *input*, divide em três e coloca cada parte no arg1, arg2 e arg3, respetivamente. (O nome dos artigos não pode conter espaços).
- **int codigoproduto()** : retorna o código ao produto, quando é inserido.
- **int produtos(char *nome)** : recebe o nome do produto, coloca-o no ficheiro “strings” e retorna a posição.
- **void insereArtigo(char *nome, char *preco)** : insere o artigo no ficheiro “artigos”.
- **void modificaNome(char *codigo, char *nome)** : escreve o novo nome no ficheiro “strings” e altera a posição no artigo correspondente ao código que recebe.
- **void modificaPreco(char *codigo, char *preco)** : modifica o preço do artigo correspondente ao código que recebe. Altera o ficheiro “artigos” e escreve no ficheiro “tmp” para guardar os artigos que têm o preço alterado.

Servidor

Quando o servidor é iniciado, este compacta o ficheiro “strings”, caso seja necessário.

Depois, cria um processo. Se for o processo filho, lê do *fifo_server* e utiliza a função **trataCliente(buf,&arg1,&arg2)** para verificar se lê só o código do artigo ou se também recebe a quantidade e retorna 1 ou 2. A seguir, verifica se já existe o preço desse artigo no *array* que guarda os preços. Depois chama a função **char* info_1 (int cod, float prec)** ou **char * info_2 (int cod, char *arg2, float prec)** conforme o valor retornado no *trataCliente*. As funções retornam uma *string* que vai ser escrita no *fifo_client*. No caso de ser o processo pai, fica a ler do *stdin* e quando ler “agregador” faz a agregação do ficheiro “vendas”. Termina quando ler EOF.

Cliente

O cliente lê do *stdin* para um *buffer*, escreve esse buffer para o *fifo_server*, depois lê a resposta do servidor a partir do *fifo_client*. O cliente termina quando lê EOF.

Agregador

Determina quantas linhas tem o ficheiro “vendas” e depois lê de um ficheiro “lidos.txt” quantas linhas já foram lidas do ficheiro “vendas”. Se o ficheiro ainda não existir é porque ainda não se agregou nenhuma vez o ficheiro “vendas”; se existir, lê o valor e começa a ler o ficheiro “vendas” a partir dessa linha. À medida que lê uma venda, escreve no ficheiro final o resultado da agregação. O nome do ficheiro da agregação é gerado através da função **nomefich(char **buffer)**.

Alguns aspetos valorizados

Nesta parte, só fizemos a parte do *caching* de preços e a compactação do ficheiro “strings”.

Para o *caching* de preços criamos, no servidor, um *array* com o tamanho correspondente ao número de artigos existentes. Depois, quando o cliente faz um pedido, primeiro verifica se o preço desse artigo está no *array*. Caso não esteja, vai ler o ficheiro “artigos” para ter acesso ao preço; caso esteja, verifica se não existe um ficheiro temporário onde tem os preços que foram modificados. Depois de retificar esses preços, elimina esse ficheiro.

Para fazer a compactação, começamos por ver quantos artigos existem e quantas linhas tem o ficheiro “strings”. Se o número de linhas for inferior a 20%, o programa imprime no ecrã que não é necessário fazer a compactação. Caso seja igual ou superior, começa a ler do ficheiro “artigos” a posição do nome no ficheiro “strings”, escreve num ficheiro temporário o nome e altera no ficheiro “artigos”. No final, move o ficheiro temporário para o ficheiro “strings”.

Conclusão

Com a realização deste trabalho, aprofundamos os vários temas lecionados nas aulas práticas como, por exemplo, o acesso a ficheiros, gestão de processos, *pipes* com nome, entre outros.

Tivemos alguns problemas com a criação do cliente, pois não o conseguíamos terminar com o EOF, mas esse problema foi ultrapassado. Também tivemos dificuldades com o *caching* de preços, que achamos que deve ser melhorado.

Por fim, não conseguimos pôr a funcionar a agregação concorrente, mas é algo que gostaríamos de fazer e acrescentar no nosso trabalho.