



www.uevora.pt
Universidade de Évora

Honesto Estudo com Longa Experiência Misturado

Departamento de Informática

Redes de Computadores

Cliente RC-Compras

- Segundo Trabalho Prático -

LUÍS POLHA, Nº 20464

MARLENE OLIVEIRA, Nº 25999

Introdução

No âmbito da disciplina de Redes de Computadores foi-nos solicitada a elaboração de uma aplicação que permitirá ao utilizador efectuar compras *online*.

A aplicação elaborada irá ligar-se a uma Central e às Lojas cuja informação será fornecida por essa mesma Central através do protocolo TCP. O protocolo TCP (*Transmission Control Protocol*) é um protocolo da camada de transporte do Modelo OSI, sendo um dos protocolos mais utilizados devido à fiabilidade na entrega dos dados entre dois elementos de uma rede. Este protocolo permite efectuar uma ligação ponto-a-ponto (entre dois pontos na rede), sendo enviado um pedido de ligação para a máquina à qual se pretende ligar. Depois de estabelecida a ligação, é possível a transferência de dados simultaneamente entre o Cliente e o Servidor durante todo o tempo em que a ligação estiver activa^[1]. O TCP faz ainda o controlo de fluxo, garante que todos os pacotes foram bem recebidos no final da transmissão e efectua a reconstrução do fluxo de dados aquando da recepção de informação, garantido que os dados enviados são recebidos pelo destinatário ordenadamente.

A aplicação elaborada irá ligar-se ao Banco através do protocolo UDP. O protocolo UDP (*User Datagram Protocol*) é um protocolo pertencente à camada de transporte do Modelo OSI, sendo um protocolo bastante simples e que permite às aplicações o acesso ao serviço de entrega de mensagens independentes (datagramas). O protocolo UDP não é tão fiável como o protocolo TCP, uma vez que não existe qualquer forma de saber se os pacotes enviados através deste protocolo foram entregues no destino ou se se perderam. A entrega de pacotes ao destinatário pode ser desordenada, sendo bastante provável que exista perda de pacotes. Ao contrário do protocolo TCP, este protocolo não possui mecanismos para iniciar, manter e encerrar uma ligação, sendo por isso mais eficiente.

A aplicação irá utilizar as funções *rc-net-lib*, que permitirão efectuar as ligações necessárias, enviar dados e receber dados.

Juntamente com a aplicação foi também desenvolvido um *makefile*, que permite gerar os ficheiros que produzem a aplicação.

[1] – activa neste sentido significa que a ligação não foi terminada.

Funcionamento da Aplicação RC-Compras

Quando o utilizador inicia a aplicação, ser-lhe-á apresentado um pequeno menu que lhe permitirá escolher se pretende iniciar sessão. Este menu será semelhante ao seguinte:



```
Deseja iniciar sessão? s = Sim/ n = Não
|
```

Figura 1 – Menu apresentado antes da aplicação iniciar a sessão do utilizador.

Se o utilizador decidir sair, o programa termina. Caso contrário, a aplicação irá estabelecer uma ligação com a Central e solicitar a identificação (que consiste numa sequência RCXXXXXX em que os X corresponderão a inteiros, por exemplo: RC12345) do utilizador e envia a mesma à Central. Caso ocorra um erro ao estabelecer a ligação com a Central, a aplicação apresentará uma mensagem de erro. Se a aplicação não enviar a identificação do utilizador à central num intervalo de tempo de dois segundos, esta irá terminar a ligação. Se a informação do utilizador for enviada correctamente, a aplicação escreve no output uma mensagem que indica que a operação foi realizada com sucesso, caso contrário também é exibida uma mensagem que informa o utilizador de que ocorreu um erro ao enviar a informação do mesmo.

Depois da informação do utilizador ter sido enviada com sucesso, a aplicação irá receber o número de lojas activas e escrevê-lo-á no output.

Seguidamente a aplicação irá receber a informação relativa às lojas activas. Para armazenar a informação das lojas foi criada uma estrutura que contém os seguintes atributos:

- um inteiro onde será armazenado o valor correspondente ao comprimento do nome da loja;
- uma string onde será guardado o nome da loja;
- um inteiro onde será armazenado o valor correspondente ao comprimento do endereço de IP da loja;
- uma string onde será armazenado o endereço de IP da loja (este endereço corresponderá a uma sequência na forma n.n.n.n e tem um comprimento máximo de quinze caracteres);
- um inteiro que corresponderá ao número da porta da loja.
- um inteiro correspondente ao número de produtos disponíveis na loja.
- um *array* de estruturas do tipo PRODUTO, que irá conter os dados relativos aos produtos que existem na loja.

Note-se que cada uma destas estruturas simboliza uma loja.

Todos os inteiros recebidos pela aplicação serão convertidos de *network byte order* para *host byte order* e serão seguidamente armazenados nos atributos correspondentes da estrutura criada. As strings que contêm o nome da loja e o endereço de IP serão inicialmente colocados em strings temporárias, que serão depois copiadas para as strings da estrutura criada.

A aplicação irá apresentar ao utilizador a informação relativa às lojas disponíveis. Para tal, foi criada a função *printInfoLoja* que recebe como argumentos o número de lojas activas e o *array* que contém as lojas activas e irá imprimir no output as informações relativas às lojas.

O utilizador poderá então escolher a loja onde pretende efectuar as suas compras. A aplicação irá receber a escolha do utilizador e, depois de procurar a mesma no *array* que contém as lojas, irá ligar-se à loja através do protocolo TCP e utilizando o endereço de IP correspondente à loja, bem como o número da porta da mesma. Se ocorrer algum erro na ligação à loja, a aplicação escreve no output uma mensagem de erro.

Quando a ligação à loja for efectuada com sucesso, a aplicação irá receber o número de produtos disponíveis na loja. Este valor será armazenado no inteiro criado para o efeito na estrutura

que simboliza a loja escolhida. Seguidamente, a aplicação irá receber os dados relativos aos diferentes produtos disponíveis. Para armazenar esta informação foi criada uma estrutura que contém os seguintes atributos:

- um inteiro que irá armazenar a referência do produto;
- um inteiro que irá armazenar o comprimento do nome do produto;
- uma string que irá armazenar o nome do produto;
- um inteiro que irá armazenar o preço do produto.

Note-se que cada uma destas estruturas simboliza um produto.

A aplicação irá apresentar ao utilizador a informação relativa às lojas disponíveis. Para tal, foi criada a função *printInfoProd* que recebe como argumentos o número de produtos disponíveis e o *array* que contém a informação relativa aos produtos disponíveis e irá imprimir no output as informações relativas às lojas.

A aplicação permitirá ao utilizador escolher quais os produtos que este deseja adquirir. O utilizador poderá decidir se deseja concluir a encomenda (para tal terá de inserir o caractere '=' no menu criado para o efeito) ou se deseja consultar a lista de produtos disponíveis (para tal terá de inserir o caractere '?' no menu criado para o efeito). Para ignorar os menus onde tais solicitações são efectuadas, o utilizador terá apenas de pressionar a tecla *Enter*. A aplicação exibirá sempre o montante actual da encomenda em curso.

Quando o utilizador decidir terminar a compra, a loja enviará à aplicação o endereço de IP do Banco e a sua porta e um código que deverá ser enviado ao mesmo. Ao contrário do que acontece com as lojas e com os produtos, os dados do Banco (IP e porta) não são guardados permanentemente, uma vez que são apenas utilizados temporariamente. Para determinar o comprimento do código a ser enviado ao banco foi utilizado o valor retornado pela função *rc_recv*, que corresponde ao número de bytes recebidos. Seguidamente, a aplicação liga-se ao Banco utilizando o protocolo UDP e enviará o código que a loja forneceu.

O Banco responderá com um código, indicando se o pagamento foi ou não autorizado. Este código deverá ser enviado à loja, que responderá com uma sequência que indica se a transacção foi efectuada com sucesso ou não. No caso de ocorrerem erros quer no contacto com o banco, quer no envio ou recepção do código de e para o banco, a aplicação irá sempre exibir uma mensagem de erro no output.