

1º Trabalho Laboratorial



Mestrado Integrado em Engenharia Informática e Computação

Redes de Computadores

Grupo:

Cláudia Margarida da Rocha Marinho - up201404493

Rui Miguel Oliveira - up201000619

Vinicius Ferretti da Silva – up201600721

Novembro de 2016

Sumário

Neste relatório será apresentado o resultado obtido do desenvolvimento de uma aplicação de protocolos de transferência de dados através de uma porta RS232 que será responsável por enviar um ficheiro de um computador (transmissor) para outro (receptor) e sem apresentar erros. Caso os dados não tenham sido recebidos devidamente, o transmissor tentará enviar novamente e o receptor só conseguirá ler os dados quando estes forem recebido sem erros.

O trabalho foi concluído com sucesso, tendo sido possível transferir dados de um computador para o outro sem erros, mesmo que a conexão fosse interrompida, ou quando erros fossem introduzidos no processo.

Introdução

Este trabalho tem como objetivo implementar um protocolo de ligações de dados e testá-lo com a transferência de ficheiros entre computadores utilizando o cabo RS232. Esta porta série foi usada para nos ajudar a compreender como ocorre a transferência de dados a baixo nível. Para este fim, além de ter sido implementada uma aplicação que permite a transferência de dados entre computadores, também foi implementada um mecanismo de detecção e controlo de erros, de forma a que a receção de ficheiros seja bem sucedida mesmo que a conexão seja interrompida.

Nas seguintes secções, vai ser encontrada informação sobre a arquitetura do projeto, a estrutura do código, os casos de uso principais, o protocolo de ligação lógica (identificação dos principais aspetos funcionais), o protocolo de aplicação, a validação (descrição dos testes realizados) e os elementos de valorização.

Arquitetura

As funções essenciais do nosso projeto foram implementadas em dois blocos: DataLink e AppLayer. No bloco DataLink, foi implementado o protocolo de ligação de dados com funções responsáveis pelo estabelecimento e terminação de ligações e com funções que garantem a transmissão sem erros de informação. O mecanismo de *stuffing* também foi implementado neste bloco devido à necessidade de transparência. No bloco AppLayer, foram implementadas funções responsáveis pelo envio e recepção dos pacotes de controlo e de dados, tendo sido usadas as funções do bloco de protocolo de ligação de dados para este efeito.

Quanto à interface, o utilizador pode escolher o modo em que quer executar o programa (receptor/transmissor) e o porte serial que quer usar.

Estrutura do código

As principais funções usadas no protocolo de ligação de dados são:

```
int llopen(int fd, int type);  
int llwrite(int fd, const unsigned char* buffer, int length);  
int llread(int fd, unsigned char* buffer);  
int llclose (int fd, int type);  
int sendPacket(int fd, char * buffer, int length);  
int receivePacket(int fd, unsigned char ** buffer, unsigned int * buffSize);
```

Temos ficheiros devidamente separados para melhor compreensão e visualização dos códigos:

- *alarm.c* e *alarm.h* - responsável pela gestão de alarme;
- *AppLayer.c* e *AppLayer.h* - contêm todo o código que diz a respeito da camada de ligação de dados;
- *Datalink.c* e *Datalink.h* - contêm todo o código pertencente a camada de ligação de dados.

Casos de uso principais

O trabalho laboratorial tem dois principais e diferentes casos de uso, para tornar as funcionalidades de uma forma mais simples ao usuário, para isso foi criada duas funções, uma para a transmissão (transmissor) de um ficheiro e outro para recebê-lo (receptor).

- Estabelecer ligação;
- Envio de dados pelo transmissor;
- Recepção de dados pelo receptor;
- Fecho dos ficheiros abertos para a realização da transferência;
- Terminar ligação;

Mais detalhes: ([VER CÓDIGO](#))

Protocolo de ligação lógica

O protocolo tem como objetivo estabelecer uma ligação confiável entre os computadores através de um cabo serial RS232 utilizado neste trabalho, sendo o mesmo implementado no DataLink.

Aspectos Fundamentais:

- Tratar das configurações da porta série e guardar as informações;
- Estabelecer e finalizar uma ligação com a porta série;
- Envio de tramas através da porta série;
- Sincronismo das tramas e processos de Stuffing e Destuffing;
- Controle de erros e de fluxo;

Aspectos implementados no DataLink:

A função `llopen()` é responsável por estabelecer a ligação lógica entre os computadores através da porta série e a função `llclose()` vai ser responsável pela fecho desta ligação. O transmissor envia um SET ao receptor e aguarda a resposta do mesmo, a trama UA. Caso isso ocorra com sucesso o programa continua em execução e se o transmissor não receber nenhuma resposta ele volta a enviar o SET novamente com o apoio da função `alarme`, e se ainda não obtiver sucesso, ele reenvia o SET no limite de vezes estabelecido pelo utilizador. (VER CÓDIGO)

A função `llwrite()` é utilizada pelo transmissor e é responsável pelo envio de tramas de informação. Encapsulamento dos dados, stuffing e envio. (VER CÓDIGO)

A função `llread()` irá ler a trama de informações enviada pelo transmissor e após recebê-la será feito o destuffing. A trama será processada pela máquina de estados que sempre voltará ao seu estado inicial quando receber um byte inesperado, irá verificar a integridade da informação e retornar uma resposta ao transmissor de acordo com a validade da trama. A resposta a ser enviado ao transmissor pode ser RR se a trama for válida ou repetida e REJ se nenhuma das anteriores ocorrer. (VER CÓDIGO)

A função `llclose()` terminará a ligação entre os dois computadores (transmissor e receptor). O transmissor enviará um DISC e espera pelo comando DISC do receptor. A ligação termina quando o receptor receber o UA, caso contrário o programa fechará com erro. (VER CÓDIGO)

Protocolo de aplicação

O protocolo de aplicação é responsável pela interação com o ficheiro a enviar/receber e com o protocolo de ligação de dados. [código: *AppLayer.h*; *Applayer.c*]

O transmissor inicia com a função `sendFile()`, invocada do `main()`, que começa por estabelecer a estrutura de termos com a invocação de `startConnection()`, e confirmar a conexão com o `llopen()`. Depois, envia um pacote de controlo do tipo START para indicar o início de envio de dados, invocando `sendControlPacket()`. Esta função cria o pacote START com as informações do ficheiro (nome e tamanho) e entrega-o ao protocolo de ligação de dados com a invocação de `llwrite()`. A seguir, começa o envio de dados através da função `sendDataPacket()`, que trata de dividir o ficheiro em pedaços mais pequenos e entregá-los a `llwrite()`. Quando termina o envio de dados, manda um pacote de controlo do tipo END com a função `sendControlPacket()`. Por fim, desliga a conexão com o `llclose()`.

O receptor tem uma estrutura quase igual ao transmissor com a óbvia diferença que recebe o ficheiro em vez de o enviar. De forma homóloga, inicia com a função `receiveFile()` e estabelece a conexão com `startConnection()` e `llopen()`. Depois cria um ficheiro onde vai guardar os dados que quer receber e invoca `llread()` dentro de um ciclo `while` até receber o pacote de controlo tipo END. O primeiro pacote que deve receber é o do tipo START, e então invoca `receiveControlPacket()` onde define o nome e tamanho do ficheiro que vai receber. A seguir recebe pacotes de dados, tendo de invocar `receiveDataPacket()` onde guarda no ficheiro os dados que recebe. Por fim, recebe um pacote do tipo END e termina com o `llclose()`.

Validação

Para a validação do código foram feitos alguns teste, foi enviado de um computador para outro um arquivo *pinguim.fig* para testar a comunicação entre a porta RS232, foi também interrompida a conexão entre eles para quando voltar o transmissor deverá enviar o ficheiro sem erros. Foi criado picos de tensão na placa para forçar erros no envio de pacotes para mostrar que o programa detecta os erros e obriga o transmissor a enviar novamente até chegar por inteiro.

Elementos de valorização

Neste projeto foram implementados os seguintes elementos de valorização:

- Implementação de REJ: REJ será uma trama enviada pelo receptor ao emissor quando os dados recebidos no receptor contém um erro no campo BCC1 e BCC2.

Conclusões

Durante o projeto foi essencial a leitura e compreensão do guião para atingirmos os objetivos propostos. Foi feita a reestruturação e organização do código para melhor entendimento e para melhor correção de erros.

Este trabalho permitiu uma melhor compreensão sobre a comunicação de dados através de um cabo serial RS232 entre dois computadores, tendo sido possível também aperfeiçoar os conhecimentos na linguagem C no processo.

Assim, pode-se concluir que o principal objetivo foi atingido com sucesso, tendo sido possível estabelecer uma ligação de dados sem erros entre dois computadores ligados através de um cabo de série.