Ligação de dados

Redes de Computadores

1. Introdução

Este projeto foi desenvolvido no âmbito da unidade curricular de Redes de Computadores e visa a implemetação de um protocolo de ligação de dados e testar este mesmo com uma aplicação de transferência de ficheiros.

2. Arquitetura e Estrutura de código

A impelmentação do protocolo pode ser dividido em diferentes unidades lógicas cada uma independente entre si. Deste modo, temos um protocolo para a aplicação onde uma das partes, i.e. o emissor comunica com o recetor usando para tal uma interface que oferece uma abstração à camada de ligação de dados entre os dois atores.

Como foi expresso no parágrafo anterior, o código encontra-se divido de modo a proporcionar diferentes camadas de abstração, isto significa que as diferentes unidades lógicas são independentes entre si. No nosso caso, essa independência é garantida com recurso à disposição do código em diferentes ficheiros - sobretudo de *header files*, mas também com o uso da *keyword* `static` que confina uma determinada declaração somente a uma unidade lógica.

De acordo com o enunciado proposto, existem 4 funções que formam uma *API* a ser usada pelas aplicações, quer do emissor, quer do recetor. Eis os cabeçalhos dessa *API*:

```
int llopen(int port, const uint8_t endpt);
ssize_t llwrite(int fd, uint8_t *buffer, ssize_t len);
ssize_t llread(int fd, uint8_t *buffer);
int llclose(int fd);
```

`int llopen(int port, const uint8_t endpt)`

Abre o canal de comunicações fornecendo o respetivo identificador. A aplicação deve fornecer o número associado à porta série e ainda um valor de modo a identificar de que "lado" da ligação se encontra. Os valores possíveis são `REVEIVER` e `TRANSMITTER` e estão definidos no ficheiro `protocol.h`:

```
#define RECEIVER 0x01
#define TRANSMITTER 0x03
```

`ssize_t llwrite(int fd, uint8_t *buffer, ssize_t len)`

Escreve os dados contidos no `buffer` no canal de comunicações. Retorna o número de *bytes* escritos no canal, ou então um valor negativo em caso de erro.

`ssize_t llread(int fd, uint8_t *buffer)`

Lê os dados disponíveis no canal de comunicações, escrevendo-os no `buffer` passado como argumento. Retorna o valor de *bytes* lidos, ou então um valor negativo em caso de erro.

`int llclose(int fd)`

Fecha o canal de comunicações.

Opções

O protocolo permite que se configurem algumas opções (em tempo de compilação) a partir do ficheiro `makefile`, são elas:

Opção	Descrição
`TIMEOUT`	Número de segundos de espera sem resposta do recetor até se desencadear uma retransmissão
`MAX_RETRIES`	Número máximo de tentativas de retransmissões até que o emissor desista de retransmitir
`MAX_PACKET_SIZE`	Tamanho máximo, em *bytes*, para os pacotes da aplicação

4. Casos de uso principais

5. Protocolo de ligação lógica

6. Protocolo de aplicação

7. Validação

8. Eficiência de protocolo de ligação

9. Conclusões

- Miguel Boaventura Rodrigues
- Nuno Miguel Paiva de Melo e Castro