

Relatório de decisões - Backup

Estruturas

Estrutura do pacote: uint inicio / uint tam / uint seq / uint tipo
8 bits 6bits 6bits 4bits

estrutura para definir sequência de mensagens
uint client / uint server
6bits 6bits

estrutura do pacote + dados

pacote / dados / paridade
3bytes / 63bytes / 1 byte -> 67 bytes no total

timeout:

5 segundos de espera

espera uma sequência que já recebeu para mandar a próxima

backup: timeout do cliente, recuperar arquivos: timeout do servidor

timeout por setsockopt e usando timestamps

Para evitar que a placa de rede sequestre bytes específicos (0x88 e 0x81), duplicamos os dados e colocamos 0xFF após todos os bytes, retirando essa sobrecarga após receber (voltando a ter 67 bytes)

Enviamos sempre uma mensagem de 67 bytes (134 com a sobrecarga para evitar sequestros de byte) pois algumas placas de rede não enviam mensagens com menos de 14 bytes, então mandamos uma mensagem "cheia" e controlamos com o campo tamanho.

paridade par: tam - seq - tipo - dados

Enviando/recebendo vários arquivos pela biblioteca glob
MD5 implementado usando a biblioteca openssl

Comandos

- ver_comandos
- cdlocal
- cdremoto
- backup_um ou bu
- backup_varios ou bv
- restaura_um ou ru
- restaura_varios ou rv
- ls
- md5
- md5_varios ou md5v
- quit

Execução:

1. necessita biblioteca libssl-dev instalada
2. executar comando make
3. sudo ./main servidor para executar o servidor
4. sudo ./main cliente para executar cliente