MC833 - PROGRAMAÇÃO DE REDES DE COMPUTADORES

Exercício 8 - Multiplexação de Entrada e Saída

Fernando Luis de Oliveira Costa - RA: 091188
Oscar dos Santos Esgalha Neto - RA: 108231

1.

As alterações que foram feitas são as seguintes:

Cliente: Foi alterado para que o cliente continue executando enquanto existir atividade na entrada padrão, neste caso leitura do arquivo texto, ou existir atividade no socket, ou seja, leitura dos dados respondidos pelo servidor. Para tal, foi implementada a função Select abordada em aula que nos permite tratar a muliplexação de I/O.

Servidor: Foi alterado para que o servidor ao invés de executar os comandos, apenas ecoe para o cliente o que foi lido e também foi alterado a forma de escape, que antes era feito através do comando exit e foi modificado para que termine quando o cliente não tiver mais dados a serem enviados.

2.

Comparando o tempo de execução de ambos:

Novo cliente usando multiplexação de E/S através de select:

```
➜  lab8 git:(master) ✗ time ./cliente 127.0.0.1 1024 < 1mb.txt > saida.txt
./cliente 127.0.0.1 1024 < 1mb.txt > saida.txt  2.14s user 0.28s system 81% cpu 2.980 total
```

./cliente 127.0.0.1 1024 < 1mb.txt > saida.txt
2.14s user
0.28s system
81% cpu
2.980 total

Cliente antigo (que não usa multiplexação):

```
➜  lab8 git:(master) ✗ time ./cliente 127.0.0.1 1024 < 1mb.txt > saida.txt
./cliente 127.0.0.1 1024 < 1mb.txt > saida.txt  3.91s user 0.30s system 70% cpu 5.980 total
```

./cliente 127.0.0.1 1024 < 1mb.txt > saida.txt
3.91s use
0.30s system
70% cpu
5.980 total

É vantajoso utilizar o código novo, ele é executado mais rapidamente porque pode continuar lendo o arquivo e enviando novas linhas para o servidor enquanto espera receber a resposta de echo. Assim que recebe uma resposta pode escrevê-la na saída. No caso do cliente antigo é preciso esperar chegar no servidor uma linha e ela ecoar de volta para o cliente antes de poder ler a próxima.