

#### **Atividade 4: Multiplexação de Entrada/Saída**

*\*\*\*\* Data de entrega: 15 de dezembro de 2020.*

*\*\*\* Este trabalho deve ser realizado utilizando linguagem C ou C++. Trabalhos que utilizarem bibliotecas externas poderão receber nota zero na implementação, pois todos os trabalhos serão compilados utilizando o padrão destas linguagens. Isto vale para quem utiliza Windows, certifique-se de compilar e executar em máquinas com Linux/Unix antes de entregar a tarefa.*

***\*\* Obrigatório a entrega de instruções de compilação e execução dos algoritmos. Opcional o envio de makefile.***

*\*\*\* Detalhes da implementação: O código não pode apresentar nenhum warning quando compilado. Para cada warning exibido na compilação será descontado 10% da nota deste trabalho. (Certifique-se de compilar seu código com a flag -Wall do gcc).*

*\*\*\* Entregáveis: **relatório e códigos**. Regra para atribuição de nota: Relatório 50% e Código 50%. Relatório arquivo deverá ser em formato PDF e entregue juntamente com os algoritmos. Não serão aceitos executáveis.*

#### **Exercício**

1. Modifique o programa cliente da atividade 2 para que este receba como entrada e envie ao servidor linhas de um arquivo texto qualquer (O arquivo será passado utilizando o caracter de redirecionamento '<').

- O cliente continuará recebendo o eco enviado pelo servidor, que deverá ser escrito em um arquivo (O arquivo será criado utilizando o caracter de redirecionamento '>'). Seu programa deverá necessariamente utilizar ou a função `select` ou a função `poll`.
- Cada linha deve ser enviada separadamente para o servidor. O servidor irá enviá-las de volta para o cliente, então cuidado com os '\n' e '\t'.
- O cliente deve finalizar sua execução assim que tiver recebido **todo o arquivo** ecoado pelo servidor.

Dica: Uma forma de verificar se o seu programa está enviando e recebendo o texto da forma correta é executar o comando `diff` no cliente entre o arquivo de entrada (passado pelo '<') e o arquivo de saída (gerado pelo '>').

**\*IMPORTANTE\*:** O código será testado com arquivos texto de cerca de 20000 contendo podendo conter palavras com acento e caracteres especiais. Para certificar que seu código está funcionando corretamente, use arquivos com tamanhos próximos a esses durante os testes.

Dica 1: Uma boa ajuda para desenvolver esta atividade poderá ser encontrada nos exemplos do livro-texto da disciplina (**capítulo 6**) e nos programas utilizados na segunda atividade parte 1 e parte. Links úteis: [select](#) e [poll](#)

Dica 2: Como teste poderá ser utilizado os arquivos disponibilizados em <https://www.gutenberg.org/>

2. Cite um exemplo para uso de multiplexação de entrada e saída.