

# MC833 - Exercício 5

Felipe Santana Dias - 215775

## 1. Explique o que é multiplexação de E/S e faça um resumo das diferenças entre os 5 tipos de E/S.

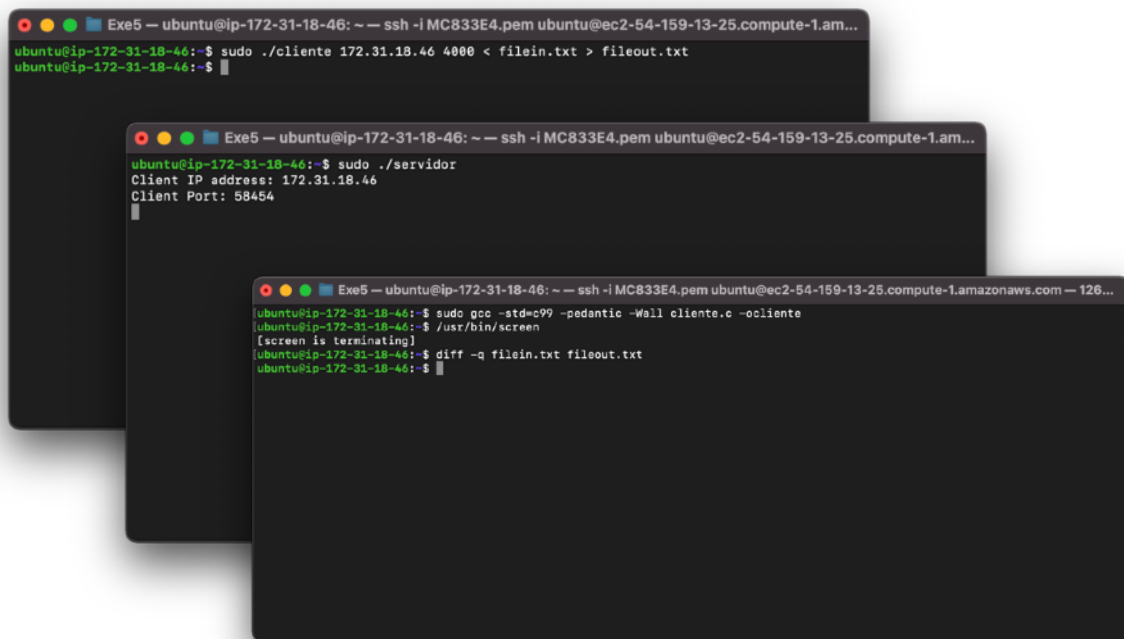
A multiplexação de E/S permite que o servidor ou o cliente receba informações do kernel à respeito das condições de entrada e saída - seja a disponibilidade de dados para leitura ou quando é necessário lidar com várias fontes de entrada simultaneamente. Essa multiplexação pode ocorrer de 5 maneiras:

1. E/S bloqueante - Ao requisitar ao kernel as informações de entrada, a aplicação fica bloqueada aguardando a resposta afirmativa dos dados (em estado de sleep), continuando suas operações apenas após receber as cópias completas dos dados.
2. E/S não bloqueante - Ao requisitar as informações de entrada ao kernel, a aplicação aguarda tanto uma resposta positiva (o que finalizaria a solicitação, recebendo o conjunto de dados), quanto uma resposta negativa (representada pelo código de erro EWOULDBLOCK). Ao receber esse código, a aplicação envia novamente uma solicitação para o kernel até que a resposta seja positiva, não existindo estado de sleep.
3. Multiplexação de E/S - Nesse caso, a solicitação de dados ao kernel é feita por alguma função (select, por exemplo) que também bloqueia a aplicação até que algum dado esteja disponível. Quando a resposta é positiva, aí então é chamada a função recvfrom para receber a cópia dos dados. Nesse meio tempo, a aplicação é novamente bloqueada.
4. E/S orientada a sinal - Diferente dos casos anteriores, essa maneira não é bloqueante da aplicação. Inicialmente é solicitado ao kernel que notifique a aplicação através do sinal SIGIO que existem dados disponíveis. Durante esse período, o processo executa normalmente até que o Signal handler receba a mensagem. Com o recebimento desse sinal, a função recvfrom é chamada para realizar a cópia dos dados, bloqueando a aplicação durante esse período.
5. E/S assíncrona - O último caso permite que o processo seja executado livremente durante todo o tempo. Ao iniciar, a aplicação sinaliza ao kernel o início da operação e continua sua execução até que o kernel forneça através de um sinal que toda a operação foi completa incluindo a cópia das informações.

## 2. Modifique o programa cliente da atividade 3.2 para que este receba como entrada e envie ao servidor linhas de um arquivo texto qualquer. Explique no relatório as mudanças que considera mais importantes para sua solução.

Para a implantação desse novo algoritmo, a diferença crucial foi no tratamento das situações em que não recebíamos dados do kernel, em que era necessário analisar cada caso para poder executar o programa.

Contudo, ao utilizar a função poll esse tratamento já é realizado, bloqueando a execução do cliente enquanto os dados não são recebidos. Essa característica foi perceptível ao executar o programa sem enviar o texto pelo caracter de redirecionamento.



Na imagem 1: Cliente após execução;  
Na imagem 2: servidor após finalização da execução do cliente;  
Na imagem 3: comparação entre os arquivos gerados utilizando a função diff.  
O parâmetro -q e a ausência de resposta significa que os arquivos são iguais.

**3. Segundo o observado na solução do ponto 2, explique qual a limitação de usar a função close para fechar a conexão e qual a vantagem de usar a função shutdown no seu lugar.**

Ao utilizar a função close, além de interromper as comunicações de envio e recebimento, também ocorre o encerramento completo do socket, implicando na possível perda de pacotes que não chegaram a tempo antes da finalização através dessa função.

A função shutdown, por sua vez, também pode interromper as comunicações de envio e recebimento, contudo o socket ainda estará habilitado para receber esses pacotes pendentes, evitando a perda de informações.