MC833 - Exercício 4

Felipe Santana Dias - 215775

1. **Adicione a função sleep no servidor.c da atividade prática anterior antes do socket ser fechado close(connfd) de modo que o servidor "segure" a conexão do primeiro cliente que se conectar. Com essa modificação, o servidor aceita a conexão de dois clientes de forma concorrente? Comprove sua resposta através de testes.**

   Ao adicionar um sleep de 30s, vemos que o servidor não aceita conexão de dois clientes de forma concorrente. Enquanto o primeiro cliente ainda está segurando a conexão, o segundo não consegue se conectar.

   Para testar essa situação, o servidor foi inicializado e dois clientes foram rodados de forma concorrente. O que se vê é que enquanto o primeiro estabelece a conexão e espera para ser encerrado, o segundo não consegue se conectar, precisando esperar o primeiro ser encerrado.

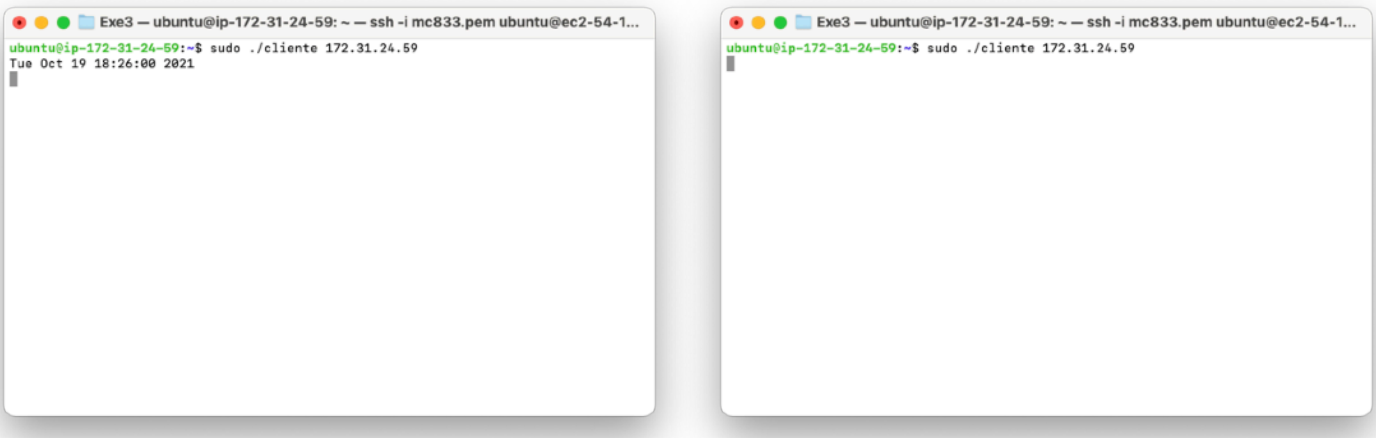


Imagem 1: cliente e servidor com modo sleep

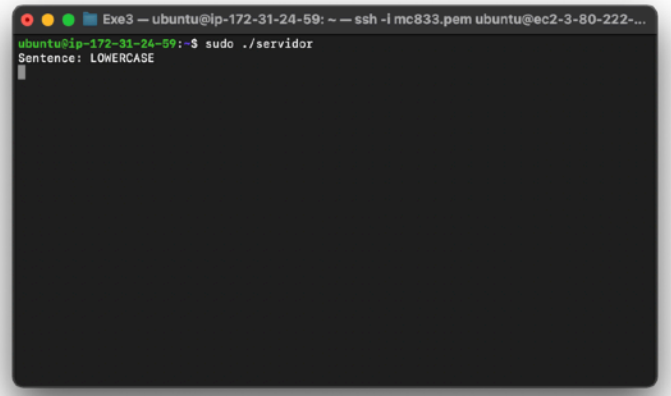
2. **Video códigos *servidorQ2.c* e *clienteQ2.c***

   Os programas cliente e servidor foram feitos de maneira que o cliente estabeleça uma conexão com o servidor, receba uma mensagem do servidor (nesse caso a mensagem era 'lowercase'), transforme essa mensagem em letras maiúsculas ('LOWERCASE') e envie essa mensagem de volta para o servidor.

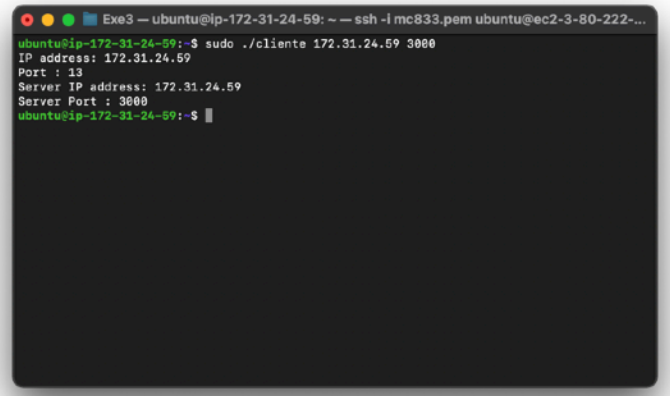
   A primeira execução foi para averiguar o funcionamento das conexões e da manipulação da mensagem, o que foi atingido com sucesso como pode ser visto na imagem 2. Para o servidor descobrir o endereço IP e a porta utilizada pelo cliente, foram testados 3 caminhos:

   1. Através da função *getnameinfo*, mas infelizmente não foi possível executar esse comando por conta de *warnings*.
   2. A segunda opção foi enviar como mensagem do cliente para o servidor através das função *send* e *recv*. Devido algumas dificuldades com a questão de endereçamento e tamanho variável dos valores, as informações conseguidas não foram satisfatórias
   3. Por fim, para concluir o objetivo do programa, utilizamos como última opção a função *getpeername*, resultando na imagem 4.

Imagem 2: cliente e servidor sem função no servidor para mostrar IP e Porta do cliente



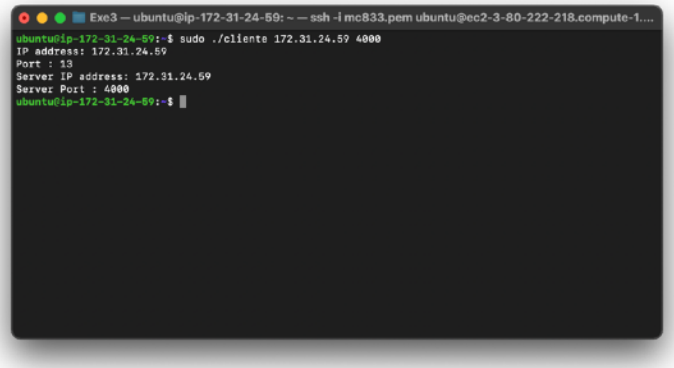
SERVIDOR



CLIENTE

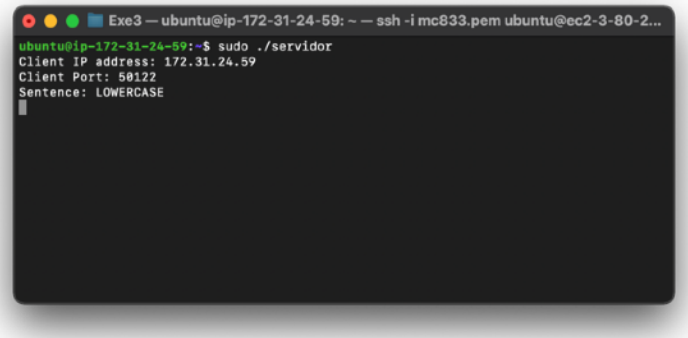
Imagem 3: cliente e servidor com 2ª opção para servidor mostrar IP e Porta do cliente

SERVIDOR



CLIENTE

Imagem 4: cliente e servidor com 3ª opção para servidor mostrar IP e Porta do cliente

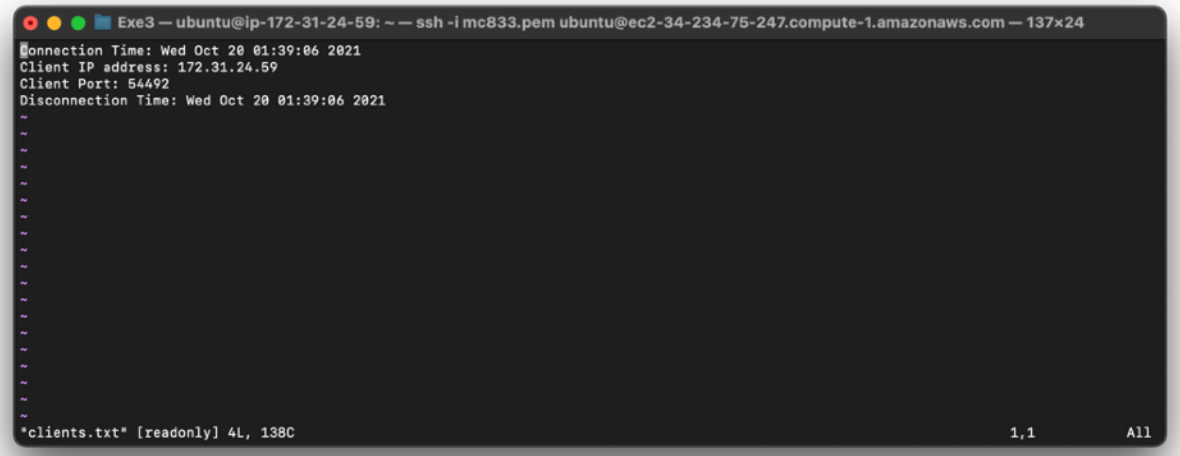


SERVIDOR

CLIENTE

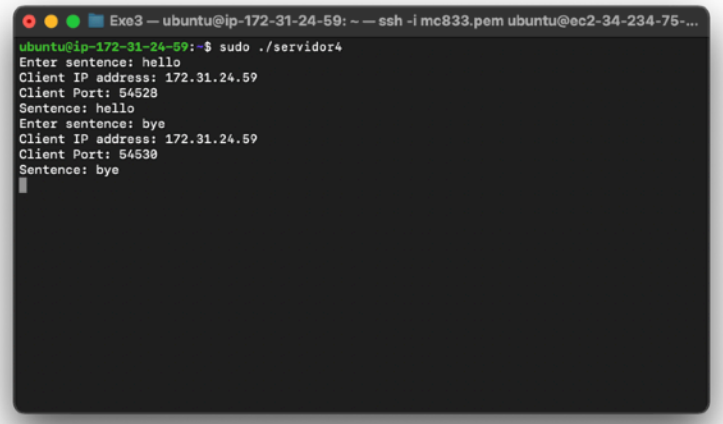
3. Video códigos *servidorQ2.c* e *clienteQ2.c*

Imagem 5: arquivo clients.txt gerado com as informações dos clientes (instante de conexão e desconexão, endereço IP e porta.

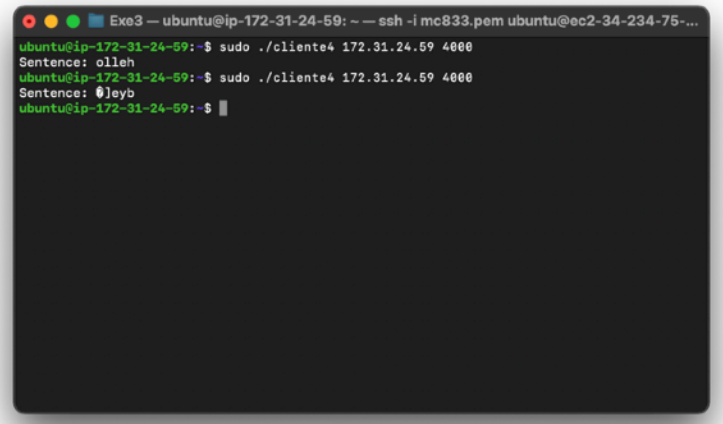


4. Video códigos *servidorQ4.c* e *clienteQ4.c*

Imagem 6: servidor e cliente invertendo palavras e encerrando com a palavra 'bye'



SERVIDOR



CLIENTE

5. Com base ainda no seu código, é correto afirmar que os clientes nunca receberão FIN neste caso já que o servidor sempre ficará escutando (LISTEN)? Justifique.

Sim, como o servidor ficar sempre escutando, quando for digitada a entrada padrão para finalizar a execução, o cliente enviará para o servidor o estado FIN e este retornará o ACK. Como a emissão da mensagem sempre parte do lado do cliente, os clientes nunca receberão FIN.