

Sistemas Operativos 2020-21

Guião Laboratorial: Comunicação entre processos usando sockets

LEIC-A / LEIC-T / LETI

IST

Este guião foca-se na comunicação entre processos usando sockets.

1. Descarregue o *zip* associado a este guião.

Este projeto implementa uma aplicação cliente-servidor muito simples, na qual o cliente envia ao servidor um nome de utilizador (do tipo *string*) e o servidor responde a esse utilizador com uma frase simpática. Os sockets usados são do domínio Unix e do tipo *datagram*.

- a. Estude o código de ambos os ficheiros. Em cada programa, identifique os momentos principais (consulte o diagrama no final deste guião):
 - i. Onde é criado o *socket* local?
 - ii. Onde é atribuído um nome ao *socket* local?
 - iii. Onde são enviadas e recebidas mensagens?
- b. Experimente compilar ambos os programas e executá-los.
Lance primeiro o servidor (indicando um nome para o *socket* servidor). Depois, noutra consola, execute o cliente (tendo o cuidado de, no 2º argumento usar o nome do *socket* servidor correto).

* * *

Nas alíneas seguintes, pretende-se estender o projeto de SO construído até agora para que o TecnicoFS passe a receber mensagens por parte de processos cliente. Para tal, é boa ideia usar o código da alínea 1 como referência.

2. Comece pelo servidor TecnicoFS.

- a. Pegando na versão atual do TecnicoFS (desenvolvida pelo seu grupo até ao momento), estenda a sua função *main* para que nela seja criado o *socket* do servidor e a ele seja atribuído um nome. Para já, assuma que o nome do *socket* é definido numa constante (i.e., usando *#define*).¹²
- b. Compile e teste a sua solução: Se os dois passos indicados acima estiverem bem implementados, conseguirá encontrar o *socket* do servidor fazendo *ls* sobre a diretoria mãe correspondente ao *pathname* do *socket*.

¹ Posteriormente, deve adaptar o programa do TecnicoFS para que o nome do *socket* seja passado por argumento de linha de comandos, tal como indicado no enunciado do 3º exercício.

² Tenha em atenção que alguns sistemas (como, por exemplo, o sigma) impõem restrições ao uso ou permissões de *sockets*. Nesses sistemas, criar o *socket* na diretoria */tmp* é uma forma simples de contornar a restrição.

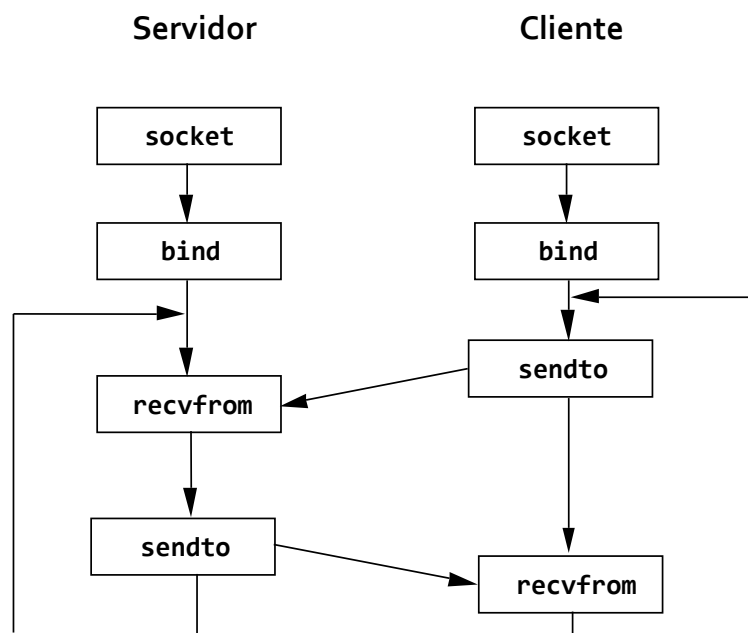
- c. Voltando ao código, acrescente agora um ciclo infinito no qual o servidor: i) espera pela receção de uma mensagem através do *socket*; ii) assumindo que a mensagem será do tipo *string*, o servidor deve imprimi-la no *stdout* após ela ser recebida. Este ciclo deve ser colocado imediatamente após a criação/preparação do *socket*. Irá testar este ciclo na alínea seguinte.
3. Construa agora um programa cliente. Para isso, descarregue o código base do 3º exercício do projeto.
- a. Implemente a função *tfsMount*. Esta deve:
 - i. Criar o *socket* do cliente e associar-lhe um nome. Nesta fase, assuma que o nome do *socket* cliente se encontra definido numa constante.³
 - ii. Guardar o nome do servidor (passado como argumento) numa variável acessível às outras funções do mesmo ficheiro de código.
 - b. Usando o mesmo procedimento da alínea 2.b, teste se o *socket* foi bem criado.
 - c. Implemente agora a operação *tfsCreate* para que envie uma *string* “*Hello world*” ao *socket* servidor.⁴ Tal como fez na alínea 2, nesta fase comece por definir o nome do *socket* servidor
4. Finalmente, experimente colocar o projeto cliente-servidor construído nas alíneas anteriores em ação.
- a. Lance o servidor.
 - b. Execute o cliente exemplo passando-lhe um ficheiro de *input* constituído apenas por comandos *create*.
 - c. Observe que as mensagens são recebidas e impressas do lado do servidor.
5. O que mudaria se pretendesse que o projeto usasse *sockets stream* em vez de *datagram*? Discuta e esboce (sem implementar) as principais alterações. Como ajuda, consulte o diagrama no final deste guião.
6. Agora aproveite o projeto construído nas alíneas 2-3, modificando e estendendo-o para resolver o 3º exercício do projeto!

³ Posteriormente, estude (ou discuta com o seu docente de laboratório) soluções mais flexíveis.

⁴ Obviamente, não é este comportamento que está especificado no enunciado do 3º exercício; a implementação da função deve ser modificada e completada depois deste guião.

Sockets

Sem Ligação (*datagram*)



Sockets

Com Ligação (*stream*)

