1. Numa máquina Linux/Unix, descarregue e descomprima o [código fonte de um servidor do Jogo do Galo/Tic Tac Toe, baseado em Sun RPC](http://disciplinas.tecnico.ulisboa.pt/leic-sod/2016-2017/labs/03-rpc/ttt.zip).  
   Sugestões:
   * Caso não disponha de máquina Unix/Linux, use a sua conta no sigma.tecnico.ulisboa.pt.  
     Pode usar um dos vários editores de texto no terminal: nano, vi, emacs, pico, joe, ...
   * Para evitar conflitos com outros RPC na mesma máquina deverá editar o identificador do programa (ttt.x) e substituir os últimos 5 dígitos pelo seu número de aluno
   * Se quiser correr o laboratório no seu próprio computador é necessário instalar os seguintes pacotes Linux (os nomes exactos variam conforme a distribuição usada): rpcbind e nfs-kernel-server.   
     Em algumas máquinas, que recorrem ao rpcbind, pode ser necessário lancar esta aplicação usando rpcbind -wi ou então usando privilégios de administração (sudo) no caso de obter a seguinte mensagem de erro: *"Cannot register service: RPC: Authentication error; why = Client credential too weak"*
2. Estude a interface remota deste servidor (ficheiro ttt.x) e explore a implementação do servidor para compreender como cada procedimento remoto está implementado (restantes ficheiros iniciados em ttt).
   * Para que serve o *mutex* usado no ficheiro ttt\_lib.c?
3. Execute na consola:  
   > rpcgen ttt.x  
   para que o SUN RPC gere os vários ficheiros de suporte à aplicação distribuída.  
   Depois execute:   
   > rpcgen -Sc ttt.x > ttt\_client.c  
   para gerar um exemplo de um cliente. (existe também a opção -Ss para gerar um exemplo de servidor)
4. Estude os ficheiros gerados.
   * Onde estão as rotinas de conversão de tipos de dados?
   * Onde é chamada a função clnt\_call()?
   * Que ficheiros pertencem ao cliente, ao servidor, e a ambos?  
     (consulte também a Makefile)
   * Que ficheiros estão incompletos e devem ser alterados pelo programador?
   * Qual o protocolo que o cliente e servidor usam para comunicar?
5. Pretende-se também que haja um processo cliente (que possivelmente corre em máquina diferente que o servidor) que interage com os jogadores e que invoca as funções do servidor remotamente, via SUN RPC.  
     
   A lógica desse cliente já está programada em local\_main.c, mas esse ficheiro está feito assumindo chamadas locais.  
   Com base no local\_main.c, modifique o ficheiro ttt\_client.c por forma a obter um cliente do servidor remoto.   
     
   *Sugestão: i) copie o código da função main do local\_main.c para dentro do método ttt\_1 de ttt\_client.c; ii) procure as linhas onde existem chamadas locais às funções currentBoard, play e checkWinner e substitua essas chamadas pelos exemplos de chamadas remotas que já existem no ficheiro ttt\_client.c gerado;* iii) Não se esqueça de verificar o retorno da chamada remota ao servidor e garantir o tratamento adequado em caso de erro.  
   Utilize o comando make para compilar.
6. Lance o servidor e experimente jogar remotamente através do cliente construído.  
   Para tal, execute na consola:  
   > ./ttt\_server &  
   (lança servidor em fundo - *background*)  
   > ./ttt\_client localhost  
   (lança cliente para jogar em servidor registado na própria máquina (localhost)
7. Pode consultar o estado dos processos com o comando ps e terminar instâncias com o comando kill.

O resto do enunciado será entregue na aula. O objectivo será estender a solução resultante do enunciado acima com mais procedimentos ou modificar alguns dos seus procedimentos actuais.

.

**Entrega da solução**

Fénix, Avaliação, Projetos, **mini Exercício 1 - Sun RPC**

A solução completa deverá ser submetida no Fénix **antes do fim da sua aula de laboratório**.  
Trabalhos submetidos depois da hora de fim da aula não serão considerados.

**Ter atenção ao seguinte:**

* Só serão aceites trabalhos de estudantes que estiveram presentes no laboratório.
* Assegure-se que a solução é enviada em formato ZIP e que não contém código compilado.  
  (faça make clean antes de zipar)
* Deverá incluir um ficheiro respostas.txt com as respostas às perguntas do enunciado do exercício.
* Deverá também incluir um ficheiro instrucoes.txt com resumo da funcionalidade implementada e com instruções para colocar o programa a funcionar como esperado.  
  Por exemplo:
  + A funcionalidade pedida foi total/parcialmente implementada **...**
  + Para compilar: make
  + O servidor deve executar com o comando: ./ttt\_server &
  + O cliente deve executar com o comando: ./ttt\_client localhost