Nome:_____

Nº ESTG: _____

Aval. Periódica - EI/Prog. Avançada

Teste Prático #2 – 2023.12.16/E

Licenciatura em
Engenharia Informática
UC de Programação Avançada
2º ano – Engenharia Informática
Regime diurno/pós-laboral
Ano letivo 2023/2024 - 1ºSemestre

Teste Prático – Enunciado E

2023.12.16 /11h15' Prova com consulta **Duração:** 90 minutos

Nome completo: _____

N.º de estudante: _____ **Regime:** [ ] Diurno [ ] Pós-laboral

IMPORTANTE

É expressamente proibido o recurso à Internet durante a prova. Qualquer utilização não autorizada da Internet leva à anulação da prova e ao reportar da situação às autoridades competentes. O mesmo sucede com outros tipos de tentativa de fraude.

- **Antes de iniciar a prova:**
  - Execute os seguintes comandos:

    `cd; mkdir -p ~/ProvaP/R_NUMERO/`

    (em que **R** deve ser substituído pela letra **D** se for do regime diurno e **N** se for aluno do regime pós-laboral e **NUMERO** deve ser substituído pelo seu número ESTG);
  - Para garantir que o seu diretório de trabalho seja o correto, faça:

    `cd ~/ProvaP/R_NUMERO/`
- **Após ter terminado a prova:**
  - Deverá proceder à criação de um arquivo TAR, fazendo uso do seguinte comando:

    `cd ~/ProvaP/R_NUMERO/; tar cvf ProvaP_YYYYMMDD_R_NUMERO.tar *`

    (em que YYYYMMDD corresponde à data corrente, e.g., 20231216, e R_NUMERO obedece ao formato acima indicado);
  - Verifique que o arquivo ".tar" que criou não está vazio, através da execução de:

    `tar tvf ProvaP_YYYYMMDD_R_NUMERO.tar`
  - Entregue o arquivo ".tar" através da plataforma moodle, no espaço reservado para o efeito. Em caso de dúvidas, pergunte ao professor;
  - Informe o professor para este validar a receção dos seus ficheiros.

Pergunta [20 valores]

(Escreva as suas respostas a esta pergunta no diretório **"~/ProvaP/R_NUMERO/Pergunta"**. Deve indicar o seu nome completo e número de estudante IPLeiria no ficheiro **README.txt** a ser criado no diretório)
NOTA 1: não é permitida a chamada a comandos externos através da função *system* ou de outra com funcionalidade similar.
NOTA 2: a solução deve ser implementada com recurso aos ficheiros do diretório **EmptyProject-client-server-template.zip**
NOTA 3: código entregue que **não compile** através do utilitário *make* e do respetivo *makefile* na máquina virtual da UC leva à atribuição da classificação de **0 (zero) valores** à resposta.

Recorrendo à linguagem C, pretende-se que implemente o serviço cliente/servidor UDP **count1bits** cujo propósito é o de contar o número de bits com valor 1 de cada um dos valores de 16 bits de um conjunto de números. O cliente deve gerar um conjunto aleatório de números de 16 bits sem sinal. O número de valores a gerar é fornecido pelo utilizador através da linha de comando. A aplicação cliente deve enviar os valores gerados num único pedido ao servidor através do protocolo de transporte UDP/IPv4. Quando recebe um pedido do cliente, o servidor deve calcular o número de bits com valor 1 de cada um dos valores recebidos. O servidor deve depois enviar a resposta para o cliente também numa única mensagem. A mensagem a enviar deve consistir num vetor de bytes, onde cada byte armazena o número de bits a 1 de cada um dos valores enviados pelo cliente.
Antes de enviar os números para o servidor, o cliente deve mostrar os valores gerados em formato hexadecimal. Quando recebe a resposta do servidor deve mostrar o número de bits a 1 de cada um dos valores enviados. Por sua vez, o servidor quando recebe o pedido do cliente deve também mostrar os valores recebidos na base hexadecimal e mostrar o número de bits a 1 de cada um dos valores recebidos depois de enviar a resposta ao cliente.
Os valores a serem trocados entre cliente e servidor devem ter em consideração a possibilidade dos sistemas terem *endianness* distintas.

Sugestão: função **srand** para inicializar o gerador de números aleatórios e **rand** para gerar um número aleatório.

Servidor: s_count1bits

O programa servidor deve suportar o seguinte parâmetro da linha de comandos, devendo o mesmo ser implementado com o utilitário gengetopt:

--port/-p <int>: porto de escuta do servidor. Deve estar compreendido entre 1 e 65535. Parâmetro obrigatório.

Cliente: c_count1bits

O programa cliente deve implementar os seguintes parâmetros da linha de comandos, devendo os mesmos serem implementados com o utilitário gengetopt:

--ip/-i <IPv4>: endereço IPv4 do servidor. Caso o valor indicado não corresponda a um endereço IPv4 válido, a aplicação termina com uma apropriada mensagem no canal de erro padrão. Parâmetro obrigatório.

--port/-p <int>: porto do servidor. Caso o valor esteja fora da gama do intervalo fechado [1, 65535], a aplicação termina com uma apropriada mensagem no canal de erro padrão. Parâmetro obrigatório.

--request_size/-r <int>: número de valores a serem gerados e enviados ao servidor. Caso o valor esteja fora da gama do intervalo fechado [1, 10], a aplicação termina com uma apropriada mensagem no canal de erro padrão. Parâmetro opcional com valor 5 por omissão.

Considere os seguintes exemplos de execução das aplicações cliente e servidor.

Exemplo 1

./s_count1bits -p 9999 Listening on UDP port: 2000 Received: 0344 59C5 022A 2650 Sent: 4 8 4 5 Received: 87E4 D3FB 37DE 461B 8AD0 Sent: 8 12 11 7 6	**./c_count1bits --ip 127.0.0.1 -p 9999 -r 4** Sending: 0344 59C5 022A 2650 Received: 4 8 4 5 **./c_count1bits --ip 127.0.0.1 -p 9999** Sending: 87E4 D3FB 37DE 461B 8AD0 Received: 8 12 11 7 6 **./c_count1bits --ip 127.0.0.1 -p 9999 -r -3** Invalid request_size. Accepted values: 1 - 10

Exemplo 2 **./s_count1bits -p 789789** ERROR: invalid port value 789789	**Exemplo 3** **./c_count1bits --ip 127.0.0.1 -p 999999 -r 7** ERROR: invalid port value 999999

	Exemplo 4 **./c_count1bits --ip 400.0.0.45 -p 9999 -r 6** ERROR: Invalid IP address '400.0.0.45'