Redes de Computadores

1st Semester 2016/2017
LEIC Alameda

Programação usando a Interface de *Sockets*

"Tradução RC"

A possibilidade de tradução de textos é crescentemente integrada em diversas aplicações. Geralmente o sistema de tradução usa uma ligação através de uma rede de comunicações para aceder ao computador que faz as traduções, que recebe um pedido de tradução e devolve o respectivo resultado. Os sistemas recentes aceitam pedidos de tradução em forma textual, mas também sob a forma de imagens de texto, devolvendo o resultado no mesmo formato (texto ou imagem) usado no pedido.

1. Introdução

O objecivo do projecto é desenvolver uma aplicação distribuída simplesque permite aos utilizadores submeter: (i) pedidos de tradução textual; ou (ii) pedidos de tradução usando uma imagem de texto, e receber de volta do servidor as respostas correspondentes.
Para este efeito, os estudantes devem desenvolver uma aplicação de utilizador e dois servidores: (i) um servidor centralizado (único) *Translation Contact Server* (TCS); e (ii) múltiplas instâncias de *Translation Servers* (TRS), cada uma destas dedicada a traduzir de uma dada linguagem para Português. A aplicação de utilizador e os servidores podem operar em máquinas distintas.
A aplicação de utilizador primeiro contacta o TCS, que opera num URL bem conhecido, perguntando a lista de linguagens disponíveis para tradução. Para cada linguagem na lista existe uma instância de um servidor TRS server (que previamente se registou junto do TCS) que responderá aos pedidos de tradução.
Quando o utilizador faz um pedido de tradução, o TCS é contactado pela aplicação de utilizador perguntando o endereço IP e o porto do servidor TRS a ser contactado para a tradução.
O servidor TRS indicado é então automaticamente contactado pela aplicação de utilizador, que identifica se se trata de um pedido de tradução de texto (*t*) ou de um ficheiro com uma imagem (*f*). Quando o TRS recebe um ficheiro com uma imagem de texto, esse ficheiro é guardado numa directoria local e a resposta é outro ficheiro com a imagem do texto traduzido, obtido também de uma directoria local.
Os identificadores das linguagens são as palavras em Português correspondentes ao nome da linguagem, tendo no máximo 20 caracteres. A directoria onde é executado o servidor TRS deve conter dois ficheiros de texto, "*text_translation.txt*" e "*file_translation.txt*".
Cada linha do ficheiro *text_translation.txt* contém (separado por espaços):

- uma primeira *string* na linguagem do TRS.
- uma segunda *string*, correspondente à tradução da primeira para Português.

Cada linha do ficheiro *file_translation.txt* file contém (separado por espaços):

- uma primeira *string* correspondente ao nome de um ficheiro com uma imagem de texto na linguagem do TRS.
- uma segunda *string* correspondente ao nome de um ficheiro com a imagem do texto traduzido para Português.

O servidor TCS manterá uma lista actualizada (eventualmente guardada num ficheiro, e.g. "*languages.txt*"), das linguagens para as quais existem servidores TRS disponíveis, bem como os endereços IP e os portos onde s respectivos servidores TCP estarão em operação. Esta informação é fornecida ao TCS por cada TRS quando inicia a sua operação.
Os estudantes devem desenvolver as três componentes do projecto: (i) servidor TCS; (ii) servidor TRS; (iii) aplicação de utilizador.
Os testes ao projecto incluirão um servidor TCS e pelo menos dois servidores TRS, que podem operar em computadores diferentes. Várias instâncias da aplicação de utilizador podem ser executadas em simultâneo. Cada servidor TRS deve conter pelo menos a tradução de 20 palavras e pelo menos 5 ficheiros com tradução de imagens de texto.

Para a implementação, os protocolos da camada de aplicação funcionarão de acordo com o paradigma cliente-servidor, usando os serviços da camada de transporte disponibilizados pela interface de *sockets*, usando os protocolos TCP e UDP.
O servidor TCS aceita pedidos de utilizadores e comunica com os servidores TRS usando UDP. O utilizador pode enviar pedidos de tradução aos servidores TRS depois de estabelecer uma sessão TCP. As respostas do TRS São igualmente enviadas à aplicação de utilizador usando a sessão TCP estabelecida.

2. Especificação do Projecto

2.1 Aplicação do Utilizador

Para invocar o programa implementando a aplicação de utilizador deve usar o comando:

`./user [-n TCSname] [-p TCSport]`,

em que:

`TCSname` é o nome da máquina onde corre o *translation contact server* (TCS). Este argumento é opcional. Se o argumento for omitido o TCS deve corer na própria máquina.

`TCSport` é o porto bem conhecido onde o servidor TCS aceita pedidos do utilizador, em UDP. Este argumento é opcional. Se omitido deve assumir o valor 58000+GN, em que GN é o número do grupo.

Depois de lançada, a aplicação espera que o utilizador indique a acção a tomar, nomeadamente:

- `list` – na sequência desta instrução a aplicação de utilizador deve contactar o TCS, usando o protocolo UDP, perguntando qual a lista de linguagens disponíveis para tradução. Em resposta o TCS indica a lista de linguagens disponíveis (*L1, L2, ... Ln*), por exemplo pela consulta de um ficheiro "languages.txt". As linguagens são mostradas ao utilizador como uma lista numerada.
- `request` *n t N* W_1 W_2 ... W_N ou
  `request` *n f filename* – na sequência desta instrução a aplicação de utilizador comunica, em UDP, com o servidor TCS, indicando o nome da linguagem desejada (L_n), pedindo o endereço IP e número de porto TCP do servidor TRS correspondente. Ao receber a resposta do TCS, a aplicação de utilizador comunica automaticamente, em TCP, com o servidor TRS indicado. Se for um pedido de tradução textual (*t*) o utilizador envia a lista de *N* palavras (W_1 W_2 ... W_N) a ser traduzidas. Se for uma tradução de ficheiro de imagem (*f*), o utilizador envia o conteúdo do ficheiro *filename*
  O servidor TRS responde com a lista de palavras traduzidas, que devem ser mostradas no écran da aplicação de utilizador, ou então a resposta será o envio de um outro ficheiro, com a imagem das palavras traduzidas, sendo o utilizador notificado quando a transferência for concluída.
  Pode assumir que num pedido de tradução – instrução `request` – não serão enviadas mais de 10 palavras.
- `exit` – a aplicação de utilizador termina a sua execução.

2.2 *Translation Contact Server* (TCS)

Para invocar o programa implementando o servidor TCS deve usar o comando:

`./TCS [-p TCSport]`,

em que:

TCSport é o porto bem conhecido onde o servidor TCS aceita pedidos, em UDP. Este argumento é opcional. Se omitido deve assumir o valor 58000+GN, em que GN é o número do grupo.

O servidor *translation contact server* (TCS) funciona no porto bem conhecido *TCSport*, suportado em UDP, e responde aos pedidos de utilizador sobre as linguagens disponíveis para tradução e onde estão disponíveis os respectivos servidores de tradução (TRS). O TCS também aceira os pedidos de registo e desregisto enviados pelos servidores TRS, guardando a informação sobre as linguagens disponíveis e os respectivos servidores TRS numa lista (eventualmente guardada num ficheiro, e.g. "*languages.txt*").
O servidor TCS lista no écran os pedidos recebidos e os endereços IP e portos originando esses pedidos.
Cada pedido recebido deve ser respondido de imediato.

2.3 Translation Server (TRS)

Para invocar o programa implementando um TRS deve usar o comando:

`./TRS language [-p TRSport] [-n TCSname] [-e TCSport],`

em que:

`language` é a linguagem a partir da qual este servidor traduz para Português.

TRSport é o porto bem conhecido onde o servidor TRS aceita pedidos do utilizador, em TCP. Este argumento é opcional. Se omitido deve assumir o valor 59000.

TCSname é o nome da máquina onde corre o *translation contact server* (TCS). Este argumento é opcional. Se o argumento for omitido o TCS deve corer na própria máquina.

TCSport é o porto bem conhecido onde o servidor TCS aceita pedidos do utilizador, em UDP. Este argumento é opcional. Se omitido deve assumir o valor 58000+GN, em que GN é o número do grupo.

O servidor TRS presta serviço ao utilizador usando o protocolo TCP, no porto *TRSport*.
O TRS aceita pedidos do utilizador para traduzir conjuntos de palavras, ou a imagem de texto incluída num ficheiro. Para as traduções o TRS consulta os ficheiros de texto, "*text_translation.txt*" e "*file_translation.txt*", como mencionado na página 1.
O servidor TRS lista no écran os pedidos recebidos e os endereços IP e portos originando esses pedidos.

3. Especificação dos Protocols de Comunicação

3.1 Protocolo *Utilizador–TCS* (em UDP)

O programa de utilizador, na sequência da instrução `list` interage com o servidor TCS, em UDP, de acordo com os seguintes pedidos e respostas:

a) `ULQ`
Após a instrução `list`, a aplicação de utilizador envia ao servidor TCS uma mensagem de pedido de listagem das linguagens disponíveis para tradução emservidores TRS.

b) `ULR` n_L *L1 L2 …* Ln_L
Em resposta ao pedido `ULQ` o servidor TCS responde, em UDP, indicando o número (n_L) e a lista de linguagens disponíveis (*L1 L2 …* Ln_L), em que*Ln* é uma palavra especificando o nome da linguagem número `n`, em Português. Estas linguagens devem ser mostradas ao utilizador como uma lista numerada.
A resposta `ULR` é enviada pelo TCS imediatamente após receber o pedido `ULQ`.
Se o pedido `ULQ` não puder ser respondido (e.g., por não haver servidores TRS disponíveis) a resposta será "`ULR EOF`". Se o pedido `ULQ` não estiver bem formulado a resposta será "`ULR ERR`".
Para simplificar, assuma queo valor máximo para n_L é 99, e que os nomes das linguagens (*Ln*) se escrevem com um máximo de 20 caracteres.

c) `UNQ` *Ln*
Após a instrução `request`, a aplicação de utilizador envia um pedido ao TCS perguntando qual o endereço do servidor TRS. Este pedido contém a identificação da linguagem desejada (*Ln*).

d) `UNR` *IPTRS portTRS*
Em resposta ao pedido `UNQ`, o servidor TCS indica o endereço IP (*IPTRS*) e o porto TCP (*portTRS*) em que o servidor TRS está disponível. A resposta `UNR` é enviada é enviada pelo TCS imediatamente após receber o pedido `UNQ`.
Se o pedido `UNQ` não puder ser respondido (e.g., nome de linguagem inválido) a resposta será "`UNR EOF`". Se o pedido `UNQ` não estiver bem formulado a resposta será "`UNR ERR`".

Em todas as mensagens acima a separação entre quaisquer dois elementos consiste num único espaço.
Cada mensagem de pedido ou de resposta termina com o caracter "`\n`".

3.2 Protocolo *Utilizador–TRS* (em TCP)

Para o utilizador pedir uma tradução (após a instrução `request`), será estabelecida uma sessão TCP com o servidor TRS seleccionado.
O protocolo de comunicação inclui os seguintes pedidos e respostas:

a) `TRQ t N` W_1 W_2 … W_N ou `TRQ f filename size data`
O utilizador pede ao TRS para traduzir palavras de texto (*t*) ou um ficheiro de imagem (*f*).
No primeiro caso (*t*), *N* especifica o número de palavras enviadas. W_1 W_2 … W_N são as N palavras enviadas, separadas por espaços. Pode-se assumir que cada palavras (*Wn*) contém no máximo 30 caracteres. Não serão enviadas mais de 10 palavras em cada instrução `request`.
No segundo caso (*f*), o ficheiro *filename*, especificado na instrução `request` é enviado. O tamanho do ficheiro *size*, em Bytes, é enviado seguido do conteúdo do ficheiro *data*.

b) `TRR t N` W_1 W_2 … W_N ou `TRR f filename size data`
Em resposta ao pedido `TRQ` o servidor TRS responde com o texto traduzido (*t*) ou um ficheiro de imagem (*f*).
No primeiro caso (*t*), *N* especifica o número de palavras enviadas. W_1 W_2 … W_N são as N palavras enviadas, separadas por espaços. Cada palavras (*Wn*) tem no máximo 30 caracteres. Não serão enviadas mais de 10 palavras.
No segundo caso (*f*), o ficheiro que contém a tradução, *filename*, é enviado. Segue-se o tamanho, *size*, do ficheiro, e o conteúdo do ficheiro, *data*.
Se o pedido `TRQ` não estiver bem formulado a resposta será "`TRR ERR`".
Se não existir tradução disponível a resposta será "`TRR NTA`".

A separação entre quaisquer dois elementos consiste num único espaço.
As mensagens terminam com o caracter "`\n`".

3.3 Protocol *TRS – TCS* (em UDP)

Quando o TRS inicia/termina a operação deve registar-se/desregistar-se junto do TCS.
A comunicação usa o protocol UDP e inclui os seguintes pedidos e respostas:

a) `SRG language IPTRS portTRS`
O TRS informa o TCS que implementa a tradução para a linguagem *language*, e opera no endereço IP (*IPTRS*) e porto TCP (*portTRS*).

b) `SRR status`
O TCS confirma (*status* = *OK*) ou recusa (*status* = *NOK*) a mensagem `SRG`.
Se existir um erro de protocolo (sintaxe) a resposta será "`SRR ERR`".

c) `SUN language IPTRS portTRS`
O TRS informa o TCS que terminou a tradução da linguagem *language*, e que *IPTRS* e *portTRS* devem ser removidos las lista de linguagens disponíveis.

d) `SUR status`
O TCS confirma (*status* = *OK*) ou recusa (*status* = *NOK*) a mensagem `SUN`.
Se existir um erro de protocolo (sintaxe) a resposta será "`SUR ERR`".

A separação entre items é um único espaço. As mensagens terminam com "`\n`".

4. Desenvolvimento

4.1 Ambiente de desenvolvimento e teste

Assegure-se que o seu código compila e executa correctamente no ambiente de desenvolvimento do laboratório LT5.

4.2 Programação

A operação dos seus programas deve basear-se no seguinte conjunto de chamadas de sistema:

- Nome do computador: `gethostname()`.
- Endereço IP do computador remoto a partir do seu nome: `gethostbyname()`.
- Servidores UDP: `socket()`, `bind()`, `close()`.
- Clientes UDP: `socket()`, `close()`.
- Comunicação UDP: `sendto()`, `recvfrom()`.
- Servidor TCP: `socket()`, `bind()`, `listen()`, `accept()`, `close()`.
- Cliente TCP: `socket()`, `connect()`, `close()`.
- Comunicação TCP: `write()`, `read()`.

4.3 Notas de implementação

O código desenvolvido deve estar adequadamente estruturado e comentado.
As chamadas de sistema `read()` e `write()` podem ler ou escrever, respectivamente, um número de *bytes* inferior ao solicitado – devem assegurar-se que a vossa implementação ainda assim funciona de forma correcta.
Tanto os processos cliente quer servidor devem terminar de forma Graciosa pelo menos nas seguintes situações de falha:

- recepção de mensagens de protocolo incorrectas;
- condições de erro das chamadas de sistema.

5 Bibliografia

- W. Richard Stevens, Unix Network Programming: Networking APIs: Sockets and XTI (Volume 1), 2nd edition, Prentice-Hall PTR, 1998, ISBN 0-13-490012-X, chap. 5.
- D. E. Comer, Computer Networks and Internets, 2nd edition, Prentice Hall, Inc, 1999, ISBN 0-13-084222-2, chap. 24.
- Michael J. Donahoo, Kenneth L. Calvert, TCP/IP Sockets in C: Practical Guide for Programmers, Morgan Kaufmann, ISBN 1558608265, 2000
- On-line manual, `man` command
- Code Complete - http://www.cc2e.com/
- http://developerweb.net/viewforum.php?id=70

6 Submissão do Projecto

6.1 Código

A submissão do projecto de incluir os ficheiros fonte dos programas desenvolvidos: *user*, *TCS server* e *TRS server*, bem como a correspondente *Makefile*.
A *makefile* deve compilar o código e colocar os executáveis na directoria de trabalho.

6.2 Ficheiros Auxiliares

A submissão do projecto deve também incluir todos os ficheiros auxiliaries necessários ara a operação do projecto (por exemplo os vários ficheiros "*text_translation.txt*" e "*file_translation.txt*").

6.3 Submissão

O projecto deve ser submetido por *e-mail* para o docente das aulas laboratoriais, **até 14 de Outubro de 2016, às 23:59**.
Deve criar um único ficheiro `zip` contendo todos o código fonte, *makefile* e todos os ficheiros auxiliares necessários à execução do projecto. O arquivo deve estar preparado para ser aberto para a directoria de trabalho e compilado com o comando `make`.
O nome do arquivo deve seguir o format: **`proj_"group number".zip`**

7 Questões

Os estudantes são encorajados a colocar as suas questões aos docentes, nomeadamente nos horários previstos para esse fim.

8 Assuntos em Aberto

Os estudantes são encorajados a pensar na forma como os protocolos considerados no projecto poderiam ser estendidos para o tornar mais genérico. Por exemplo, como poderia o utilizador contribuir com novos pares de palavras e as suas traduções, para serem incluídos em "*text_translation.txt*" e/ou em "*file_translation.txt*" de um TRS?