TÉCNICO LISBOA

Redes de Computadores

1st Semester 2015/2016
LEIC Alameda

Programming using the *Sockets* interface

"Online Questionnaire"

E-learning is becoming an increasingly important way of accessing knowledge from anywhere an Internet connection is available, allowing students to learn at their own pace and managing their schedule. Evaluation is a key component of a learning experience, with the focus of this project being the creation of a platform for supporting online questionnaires.

1. Introduction

The goal of this project is to develop an application for students to complete evaluation questionnaires online.
For this purpose, two server applications are considered: one unique centralized Evaluation Contact Point (ECP), and several instances of a Topic Evaluation Server (TES), possibly running on different machines.
The ECP, whose URL is well-known, is contacted first by the student, and provides a list of evaluation topics.
For each topic in the above referred list, there is a TES server instance that should be contacted next by the student application to obtain the corresponding questionnaire file (for instance in PDF format).
When explicitly requesting a questionnaire to the ECP, the student application specifies the topic (T_{nn}) on which to be evaluated. As a reply the ECP informs the student application of the IP address and port of the topic evaluation server (TES) to be contacted to ask for the questionnaire.
The appointed TES is then automatically contacted by the student application, which provides the student identification (*SID*) – a 5 digit number – and the selected questionnaire topic number T_{nn}. When the TES is contacted by the student, a unique transaction identifier string (*QID*) –with up to 24 characters – is assigned to this request of this student, and the TES transfers the selected questionnaire file, specifying a deadline for receiving the answer. The questionnaire file is stored into a local directory.
The topic numbers, T_{nn}, should follow the format "Tnn", with *nn* being a number between 1 and 99. The directory where the ECP runs should contain a text file, "topics.txt", where each line contains (separated by spaces):

- a string with the topic name.
- IP address of the TES containing the questionnaires for this topic.
- Port number of the TES containing the questionnaires for this topic.

All topics for this project must be related to the "Computer Networks".
Each questionnaire is stored into a single file named "TnnQFxxx.pdf", where *xxx* refers to the questionnaire number for a given topic *Tnn*.

Each questionnaire should be composed by of a set of 5 questions, including text and possibly also images. Following each question are 4 answer options (denoted: *A*, *B*, *C* and *D*), of which one and only one is correct.
The TES will also store a text file with the correct answers (one per line). The questionnaire answers file should have a name like "TnnQFxxxA.txt", where *xxx* refers to the questionnaire number for a given topic *Tnn*.
Once the student has finished answering the questionnaire he submits the answers to the TES and, as a reply, the score is returned. The TES also informs the ECP of the results of the student in this questionnaire. The ECP keeps a record of the student results.

For the implementation, the application layer protocols operate according to the client-server paradigm, using the transport layer services made available by the socket interface, using the TCP and UDP protocols.
The ECP accepts student requests and communicates with TES servers using UDP. The student requests the selected questionnaire from the TES after establishing a TCP connection. The questionnaire answers are also sent back by students to TES servers using TCP connections.

2. Project Specification

2.1 User

The program implementing the student application should be invoked using the command:

`./user SID [-n ECPname] [-p ECPport]`,

where:

SID is the student identity number, composed of 5 digits (e.g. 76543).

ECPname is the name of the machine where the central evaluation contact point (ECP) runs. This is an optional argument. If this argument is omitted, the ECP should be running on the same machine.

ECPport is the well-known port where the ECP server accepts user requests, in UDP. This is an optional argument. If omitted, it assumes the value 58000+GN, where GN is the group number.

Once the user program is running, it waits for the user to indicate the action to take, notably:

- `list` – following this instruction the user application should contact the ECP, using the UDP protocol, asking for the list of topics for which questionnaires are available in TES servers. In its reply the ECP provides the list of available questionnaire topics by checking the "topics.txt" file. These topics will be displayed to the student as a numbered list.
- `request` *Tnn* – following this instruction the student application communicates in UDP with the ECP server, providing the desired topic number (T_{nn}), asking for the IP address and TCP port number of the TES server. Once the reply from the ECP server is received, then the student application automatically communicates in TCP with the TES server, providing the student identification (*SID*) and requesting a questionnaire on the desired topic. The TES replies informing the student of the questionnaire ID (*QID*), the deadline for submitting the reply to the questionnaire, and sending the file containing the questionnaire, which should be stored in the local directory, with a name composed by concatenating the QID and the questionnaire file extension ("pdf"). When the file transfer concludes, the student should be notified by the application.
- `submit` *answers_sequence* – the student application communicates in TCP with the TES to transfer the sequence of *answers* (e.g., `submit D A C A B`), corresponding to the student answers of the questionnaire's 5 questions. Also the *SID* and *QID* are sent to the TES. The TES should then reply to the student with the questionnaire score (a negative score means the questionnaire answers were submitted after the deadline). For valid questionnaire answers, the TES will also inform the ECP of the SID, QID and score obtained.
- `exit` – the student application terminates.

2.2 *Central Evaluation Contact Point Server* (ECP)

The program implementing the *Central Evaluation Contact Point Server* should be invoked using the command:

`./ECP [-p ECPport]`,

where:

ECPport is the well-known port where the ECP server accepts user requests, in UDP. This is an optional argument. If omitted, it assumes the value 58000+GN, where GN is the number of the group.

The central evaluation contact point server (ECP) makes available a server with well-known port *ECPport*, supported in UDP, to answer student requests for questionnaire topic listing and for providing TES server identification. The ECP also collects the questionnaire results sent by TES servers, being able to produce statistics on students' performance and on the more popular questionnaire topics. These statistics are stored in the file "stats.txt".
The ECP server can answer the *TQR* messages received from students (following the *list* instruction), returning the list of questionnaire topics available after consulting the file "topics.txt". Also, following the student's *request* instruction, the ECP receives a *TER* message and returns the IP address and TCP port number of the TES server the student application should then contact to retrieve the desired questionnaire. The TES server IP and port numbers can be consulted on the file "topics.txt".

The ECP server outputs to the screen the received requests and the IP and port originating those requests.
Each user request should be responded immediately.

2.3 Topic Evaluation Server (TES)

The program implementing the *Topic Evaluation Server (TES)* should be invoked using the command:

`./TES [-p TESport] [-n ECPname] [-e ECPport]`,

where

TESport is the well-known port where the TES server accepts requests from users. This is an optional argument. If omitted, it assumes the value 59000.

ECPname is the name of the machine where the central evaluation contact point (ECP) runs. This is an optional argument. If this argument is omitted, the ECP should be running on the same machine.

ECPport is the well-known port where the ECP server accepts user requests, in UDP. This is an optional argument. If omitted, it assumes the value 58000+GN, where GN is the group number.

The TES server is supported in the TCP protocol with well-known port *TESport*.
The TES accepts user requests for the transfer of a questionnaire file, and it receives the student answers to questionnaires. Once answers are received, the TES checks them, computes the score and sends it to the student application and to the ECP.
The TES outputs to the screen the nature of the received requests and the IP and port which originated those requests.

3. Communication Protocols Specification

3.1 *User–ECP* Protocol (in UDP)

The user program, following the `list` instruction interacts with the ECP server in UDP according to the following requests and replies:

a) `TQR`
Following the `list` instruction, the student application sends a topic query request message to the ECP server, asking the list of questionnaire topics available in the TES servers.

b) `AWT` n_T *T1 T2 … Tn*$_T$
In reply to a `TQR` request the ECP server replies in UDP indicating the number (n_T) and the list of available questionnaire topics (*T1 T2 … Tn*$_T$), where *Tn* is a string of characters specifying the name of questionnaire topic number `n`. These topics will be displayed to the student as a numbered list. The reply is sent by the ECP immediately after receiving the request.
If the `TQR` request cannot be answered (e.g., no questionnaire topics available) the reply will be the string "`EOF`". If the `TQR` request is not correctly formulated the reply is the string "`ERR`".

To simplify, assume that n_T value is at most 99, and that each topic name (*Tn*) contains no more than 25 characters.

c) `TER` T_{nn}
Following the `request` instruction, the user sends a request to the ECP asking for the topic evaluation server (TES) address details. This request contains the identification of the desired questionnaire topic number (T_{nn}).

d) `AWTES IPTES portTES`
In reply to a `TER` request, the ECP server replies indicating the IP address (`IPTES`) and the TCP port number (`portTES`) where the *Topic Evaluation Server* (TES) that should be contacted to ask for the questionnaire is available. The reply is sent by the ECP immediately after receiving the request.
If the `TER` request cannot be answered (e.g., invalid topic number) the reply will be the string "`EOF`". If the `TER` request is not correctly formulated the reply is the string "`ERR`".

Each message of request or reply ends with the character "`\n`".

3.2 *User–TES* Protocol (in TCP)

For the user to receive a questionnaire file (following the `request` instruction), or to submit the answers to a questionnaire (following the `submit` instruction), a connection with the selected TES server is established in TCP.
The protocol includes the following requests and replies:

a) `RQT` *`SID`*
The student requests the TES to send one of the available questionnaires on the previously selected topic. The student ID (*SID*) is provided to the TES.

b) `AQT` *`QID time size data`*
Following a `RQT` request, a unique transaction identifier (*`QID`*) is assigned to this request (not longer than 24 characters) and a deadline *`time`* for submitting the answers is established, in the format *`DDMMMYYYY_HH:MM:SS`*, for instance *`09JAN2015_20:00:00`*. The questionnaire file `size`, in Bytes, is also sent, followed by the corresponding data. If the request is not well formulated the answer will be the string "`ERR`".

c) `RQS` *`SID QID V1 V2 V3 V4 V5`*
Following the `submit` instruction given by the student, its application sends the questionnaire answers sequence to the TES server, identifying it with the *`SID`* and the previously assigned *`QID`*. Each of the answer values (*V1* to *V5*) can take one of the values *A, B, C* or *D*. To indicate an absence of reply the value *N* should be used.

d) `AQS` *`QID score`*
Following a `RQS` request the response will contain the *`QID`* and an integer number corresponding to the *`score`*, in percentage, obtained by the student. In case the questionnaire was submitted after the deadline, then the negative number "-1" is returned. If the `SID-QID` pair does not match "-2" is returned. If the request is not well formulated the answer will be the string "`ERR`".

Each message of request or reply ends with the character "`\n`".

3.3 *TES – ECP server* Protocol (in UDP)

When the TES has computed a score for a student questionnaire (after the student uses the `submit` instruction), the ECP server is informed of the obtained score.
The communication uses the UDP protocol and includes the following requests and replies:

a) `IQR` *`SID QID topic_name score`*
The TES informs the ECP that student *`SID`* answered questionnaire *`QID`*, on *`topic_name`*, and obtained the percentage *`score`*.

b) `AWI` *`QID`*
In reply to a `IQR` request the ECP server confirms having received information about the results of questionnaire `QID`. If there is a protocol (syntax) error the answer will be "`ERR`".

Each message of request or reply ends with the character "`\n`".

4. Development

4.1 Development and test environment

Make sure your code compiles and executes correctly in the development environment available in lab LT5.

4.2 Programming

The operation of your program should be based on the following set of system calls:

- Computer name: `gethostname()`.
- Remote computer IP address from its name: `gethostbyname()`.
- UDP server management: `socket()`, `bind()`, `close()`.
- UDP client management: `socket()`, `close()`.
- UDP communication: `sendto()`, `recvfrom()`.
- TCP client management: `socket()`, `connect()`, `close()`.
- TCP server management: `socket()`, `bind()`, `listen()`, `accept()`, `close()`.
- TCP communication: `write()`, `read()`.
- Concurrent servers: `fork()`.

4.3 Implementation notes

Developed code should be adequately structured and commented.
The `read()` and `write()` system calls may read and write, respectively, a smaller number of bytes than solicited – you need to ensure that your implementation still works correctly.
Both the client and server processes should terminate gracefully at least in the following failure situations:

- wrong protocol messages received from the corresponding peer entity;
- error conditions from the system calls.

5 Bibliography

- 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 Project Submission

6.1 Code

The project submission should include the source code of the programs implementing the *user*, the *ECP server* and the *TES server*, as well as the corresponding *Makefile*.
The makefile should compile the code and place the executables in the current directory.

6.2 Questionnaire Examples

Together with the project submission you should also include at least five questionnaire examples (both the questions and the answers files), on topics related to "Computer Networks". The quality of the questionnaires will be taken into consideration when evaluating the project.

6.3 Submission

The project submission is done by e-mail to the lab teacher, **no later than October 9, 2015, at 8 PM**.
You should create a single `zip` archive containing all the source code, makefile, questionnaires and other files required for executing the applications. The archive should be prepared to be opened to the current directory and compiled with the command `make`.
The name of the archive should follow the format: **`proj_"group number".zip`**

7 Questions

You are encouraged to ask your questions to the teachers in the scheduled foreseen for that effect.