pdfauthor=Matthew Leon Dailis, pdftitle=Exercise 1: Client/Server application, pdfkeywords=, pdfsubject=, pdfcreator=Emacs 24.5.1 (Org mode 9.1.7), pdflang=English

Exercise 1: Client/Server application

Matthew Leon Dailis

March 9, 2018

Contents

1 TODO Definitions of Components

The central component of this system is the server. It contains the database, and provides a message-queue interface for clients to connect. There are arbitrarily many clients.

2 TODO Communication protocol

3 TODO Server

1. Single or multithreaded
2. Stateful or stateless

4 TODO Naming

5 Message Design

There is a single message type for both requests and responses. It has 5 fields and is structured as follows:

Field	Size
char type/status	1 byte
int key/count	4 bytes
char value1[256]	256 bytes
float value2	4 bytes
char return_queue[35]	35 bytes
Total	300 bytes

1. *char* **type/status**

(a) Requests

The type indicates the requested operation for the server. The type indicates which other fields are relevant to this message.

#	command	Explanation
0	init	This is the call to make sure that a server is up and running. All other fields are ignored.
1	set_value	Requests to store **value1** and **value2** associated with a new key, **key**.
2	get_value	Retreive the information associated with **key**.
3	modify_value	Store **value1** and **value2** under the existing key, **key**.
4	delete_key	Delete information associated with key, **key** and remove the key from the database
5	num_items	Request the number of items in the database

(b) RESPONSE: This is the status of the request

- For *init*, *set_value*, *modify_value*, and *delete_key*, this is the only relevant field in the message.

0	success
-1	error

- For *get_value*, if the **status** is 0, the requested values are stored in the **value1** and **value2** fields of the message.
- For *num_items*, if the **status** is 0, the returned count is stored in the **key/count** field of the message.

2. *int* **key/count**

   (a) Request: this is the key of the element requested

   (b) Response: (only applicable to *num_items*) The number of items in the database

3. *char*[256] **value1**

   This represents the string value that is stored or will be stored in the database.

4. *float* **value2**

   This represents the float value that is stored or will be stored in the database.

5. *char*[32] **return_queue**

   (a) Request: the message queue to which the server should write its response

   (b) Response: ignored

6 TODO My protocol

7 TODO Security

8 TODO Implementation