

Санкт-Петербургский политехнический
университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе

Дисциплина: Сети и телекоммуникационные технологии
Тема: Организация сетевого взаимодействия. Протокол TCP

Выполнил студент группы 33501/1

(подпись) У.А. Пеньевская

Преподаватель

(подпись) А.О.Алексюк

19 декабря 2017 г.

Санкт-Петербург
2017

Содержание

1	Цель работы	2
2	Индивидуальное задание	2
3	Разработанный прикладной протокол	2
3.1	Описание структуры приложения на основе TCP	3
4	Тестирование приложения на основе TCP	4
4.1	Тестирование со стороны серверного приложения	4
4.2	Тестирование со стороны клиентского приложения	5
4.3	Обработка ошибок и предупреждений клиентского приложения	9
4.3.1	Неполный список аргументов	9
4.3.2	Избыток цифр в ID	9
4.3.3	Некорректный ID	10
4.3.4	Слишком длинное название <NAME>	10
4.3.5	Слишком длинное <SHORT NAME>	10
4.3.6	Курс с таким ID не найден	10

1 Цель работы

Изучение принципов программирования сокетов с использованием протокола TCP.

2 Индивидуальное задание

Разработать распределенную систему, состоящую из приложений клиента и сервера, для распределенного выставления/просмотра курсов валют. Информационная система должна обеспечивать параллельную работу нескольких клиентов.

Основные возможности: Серверное приложение должно реализовывать следующие функции:

1. Прослушивание определенного порта
2. Обработка запросов на подключение по этому порту клиентов
3. Поддержка одновременной работы нескольких клиентов с использованием механизма нитей и средств синхронизации доступа к разделяемым между нитями ресурсам.
4. Принудительное отключение конкретного клиента
5. Добавление новой валюты (кода валюты)
6. Удаление валюты
7. Добавление курса конкретной валюты
8. Выдача пользователю списка имеющихся валют с текущими курсами и абсолютными/относительными приращениями к предыдущим значениям
9. Выдача пользователю истории курса конкретной валюты
10. *Сохранение состояния при выключении сервера

Клиентское приложение должно реализовывать следующие функции:

1. Возможность параллельной работы нескольких клиентов с одного или нескольких IP-адресов
2. Установление соединения с сервером (возможно, с регистрацией на сервере)
3. Разрыв соединения
4. Обработка ситуации отключения сервером
5. Получение и вывод списка валют с котировками/изменениями
6. Передача команды на добавление валюты
7. Передача команды на удаление валюты
8. Передача команды на добавление курса валюты
9. Получение и вывод истории котировок валюты

Настройка приложений:

Разработанное клиентское приложение должно предоставлять пользователю возможность задания IP-адреса или доменного имени сервера, а также номера порта сервера.

Тестирование: Для тестирования запускается сервер системы котировок валют и несколько клиентов. В процессе тестирования проверяются основные функциональные возможности разработанной системы.

3 Разработанный прикладной протокол

Протокол TCP имеет следующий шаблон сообщения:

<команда> <аттрибут> <аттрибут>

В начале сообщения, всегда присутствует тип команды, далее в зависимости от команды могут идти (в зависимости от типа команды) атрибуты, которые отделены друг от друга пробелом.

	Код команды	Атрибуты	Действия	Ответ сервера
1	A	<ID> <NAME> <SHORT NAME> <COURSE>	Добавление новой валюты Add New Finance Position	Added Finance Data: ID: Name: Short name: Course: absCourse: conCourse
2	D	<ID>	Удаление валюты Delete Finance Position	Finance Data: ID:0 Name: Short name: Course: 0.00 absCourse: 0.00 conCourse: 0.00
3	R	Отсутствуют	Выдача пользователю списка имеющихся валют с текущими курсами и абсолютными/относительными приращениями к предыдущим значениям Read All Finance Courcess	Finance Data: ID:0 Name: Short name: Course: 0.00 absCourse: 0.00 conCourse: 0.00
4	C	<ID> <COURSE>	Добавление курса конкретной валюты Add Finance Course	Finance Data: ID:0 Name: Short name: Course: 0.00 absCourse: 0.00 conCourse: 0.00
5	O	Отсутствуют	Разрыв соединения Exit Programm	-disconnect
6	M	Отсутствуют	Вывод меню Show Menu	-
7	H	<ID>	Получение и вывод истории котировок валюты Show History	-

Таблица 1: Команды со стороны клиента

Список команд, которыми оперирует клиент:

1	Add New Finance Position :	A <ID> <NAME> <SHORT NAME> <COURSE>
2	Delete Finance Position :	D <ID>
3	Read All Finance Courcess :	R
4	Add Finance Course :	C <ID> <COURSE>
5	Exit Programm :	O
6	Show Menu :	M
7	Show History :	H <ID>

Посылка от сервера – клиенту. Посылка состоит из символа типа char – команды, а также, при необходимости, - данных.

1	C - Client List
2	k <Socket number> - Kill klient

3.1 Описание структуры приложения на основе TCP

Сервер:

Функция main:

- Инициализация всех используемых переменных;
- Запуск событий, если таковые уже имеются;
- Создание сокета;
- Создание потока для прослушивания сокета;
- Цикл чтения команд.
- Поток для прослушивания сокета.

- Прослушивание сокета;
- Добавление подключенного клиента в список;
- Создание сокета для работы с клиентом;
- Создание потока для работы с клиентом.
- Цикл чтения команд;
- Чтение команды из стандартного ввода;
- Реакция на введенную команду (при корректном вводе команды) или игнорирование команды.
- Поток для работы с клиентом;
- Получение сообщения от клиента;
- Анализ сообщения – является ли сообщение командой;
- Ответ на команду или вывод сообщения с пометкой отправителя.

Клиент:

Функция main:

- Чтение ip адреса сервера;
- Подключение к серверу;
- Создание потока для получения данных от сервера;
- Цикл чтения данных и отправка серверу.
- Поток для получения данных от сервера;
- Побайтовое получение данных от сервера;
- Реакция на полученные данные .

4 Тестирование приложения на основе TCP

4.1 Тестирование со стороны серверного приложения

Запускаем клиент-серверное приложение

```

MSAS starting...
Opening socket...
Success!
Connecting...
Connected to 127.0.0.1
Welcome to the Finance Course programm client!

Main menu:
=====
:====Commands:=====|=====Semantic:=====
: Add New Finance Position: | A <ID> <NAME> <SHORT NAME> <COURSE>
: Delete Finance Position: | D <ID>
: Read All Finance Course: | R
: Add Finance Course: | C <ID> <COURSE>
: Exit Programm: | O
: Show Menu: | M
: Show History | H <ID>
:=====
: Enter your command... And PRAY!....
:=====

Mail Server Started...
MSAS starting...
Listening starting...
Success!
Bind starting...
Listening...
Success!
Server console
C - Client List
k <Socket number> - Kill klient

Waiting for connections...
Accepting...
+Ulya-11 [127.0.0.1] new connect!
1 user on-line
ID: 1 Addr: 127.0.0.1 S: 132
Waiting for commands...
  
```

Рис. 1: Клиент-серверное приложение "Курс валют"

Запускаем еще одного клиента

Список подключенных клиентов можно посмотреть с помощью команды - C

Отключить клиента - k <Socket number>

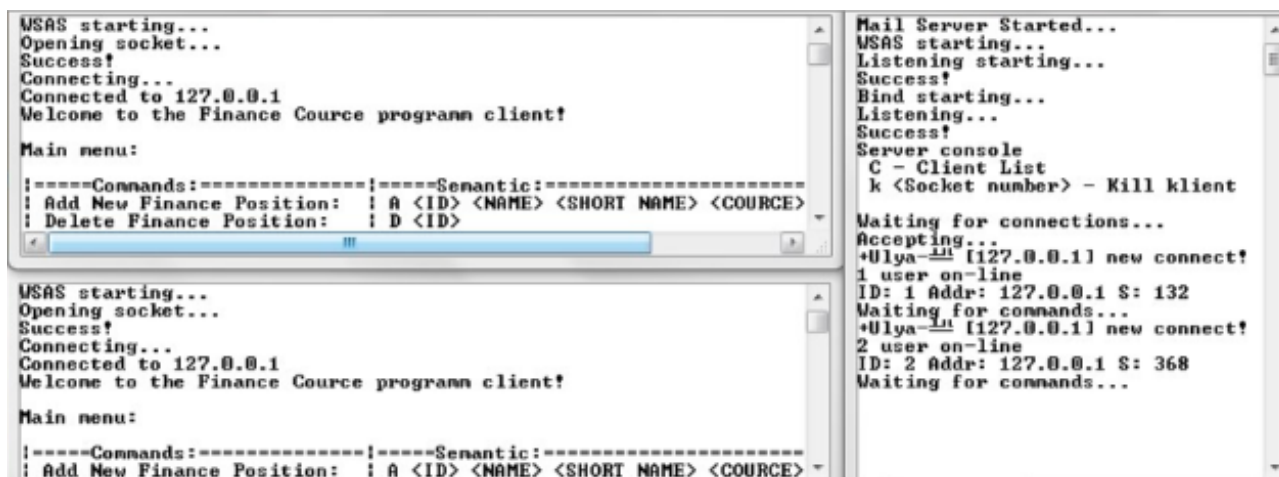


Рис. 2: Подключение второго клиента

```

C
ID: 1 Addr: 127.0.0.1 S: 132
ID: 2 Addr: 127.0.0.1 S: 368
Sucessfully
Server console
C - Client List
k <Socket number> - Kill klient
  
```

Рис. 3: Список подключенных клиентов

```

C
ID: 1 Addr: 127.0.0.1 S: 132
ID: 2 Addr: 127.0.0.1 S: 368
Sucessfully
Server console
C - Client List
k <Socket number> - Kill klient

k 132
Server console
C - Client List
k <Socket number> - Kill klient

recv failed with error: 10004
-disconnect
1 user on-line
Complited!
C
ID: 2 Addr: 127.0.0.1 S: 368
Sucessfully
Server console
C - Client List
k <Socket number> - Kill klient
  
```

Рис. 4: Отключение одного из клиентов

4.2 Тестирование со стороны клиентского приложения

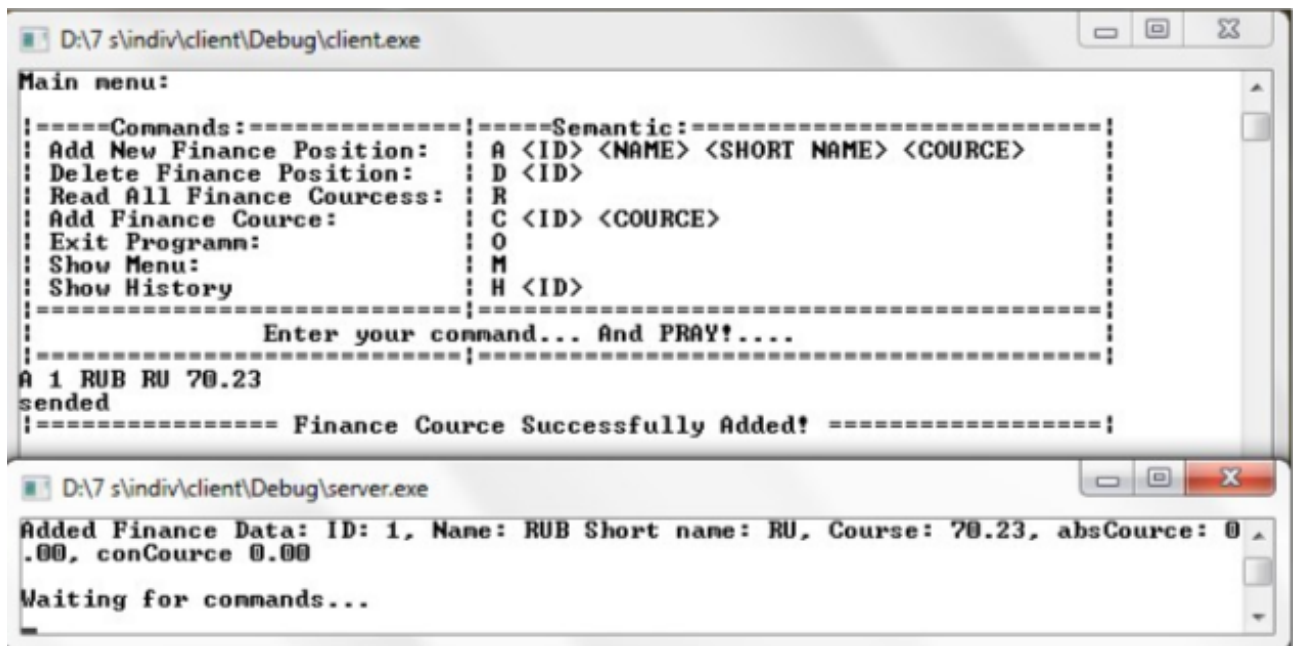
Проверка работоспособности команд

Разорвем соединение с помощью команды O (Exit program)



Рис. 5: Разрыв соединения

Добавим новую валюту вводом команды: A 1 RUB RU 70.23



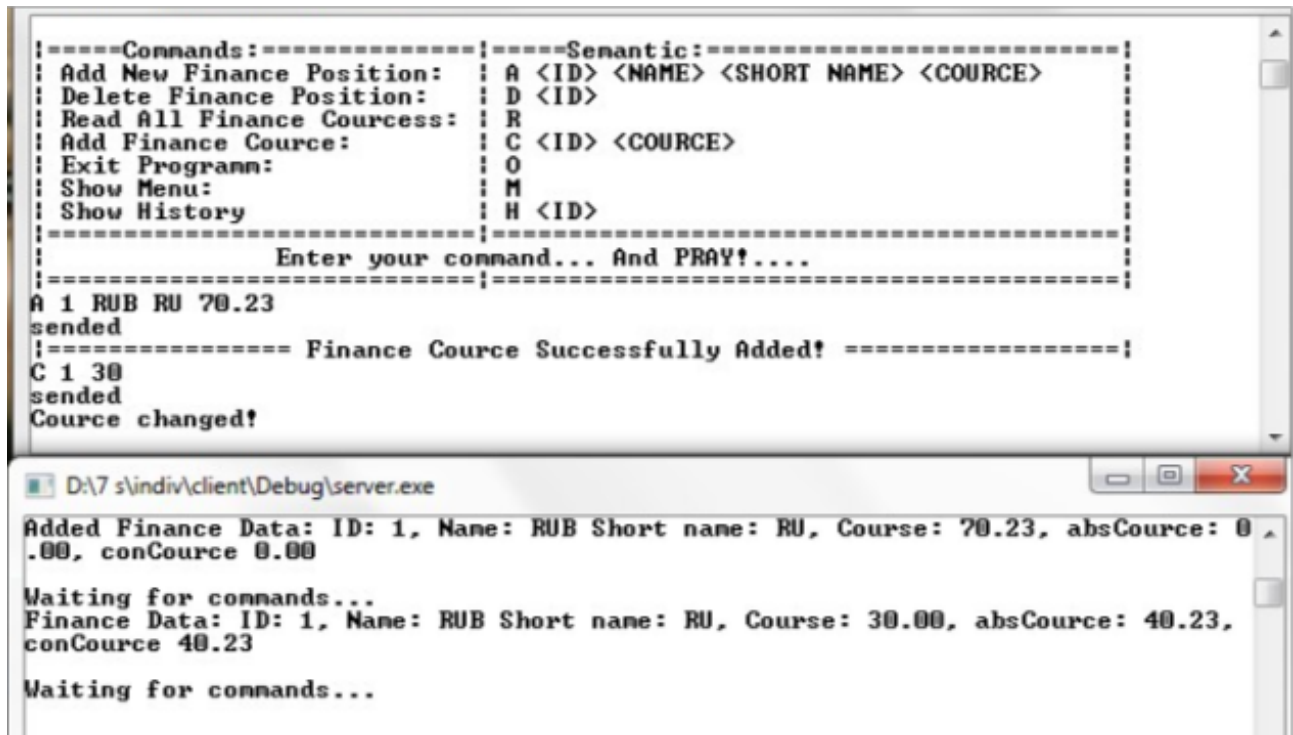
```
D:\7 s\indiv\client\Debug\client.exe
Main menu:
|-----Commands:-----|-----Semantic:-----|
| Add New Finance Position: | A <ID> <NAME> <SHORT NAME> <COURSE> |
| Delete Finance Position: | D <ID> |
| Read All Finance Courcess: | R |
| Add Finance Course: | C <ID> <COURSE> |
| Exit Programm: | O |
| Show Menu: | M |
| Show History | H <ID> |
|-----|-----|
| Enter your command... And PRAY!.... |
|-----|-----|
A 1 RUB RU 70.23
sended
|===== Finance Course Successfully Added! =====|

D:\7 s\indiv\client\Debug\server.exe
Added Finance Data: ID: 1, Name: RUB Short name: RU, Course: 70.23, absCourse: 0
.00, conCourse 0.00
Waiting for commands...
```

Рис. 6: Добавление новой валюты

Сервер выполнил команду добавление валюты (Added Finance Data)

Изменим курс валюты с помощью команды: A 1 30



```
=====Commands:=====|=====Senantic:=====|
| Add New Finance Position: | A <ID> <NAME> <SHORT NAME> <COURSE> |
| Delete Finance Position: | D <ID> |
| Read All Finance Courcess: | R |
| Add Finance Course: | C <ID> <COURSE> |
| Exit Programm: | O |
| Show Menu: | M |
| Show History | H <ID> |
|=====|=====|
| Enter your command... And PRAY!.... |
|=====|=====|
A 1 RUB RU 70.23
sended
|===== Finance Course Successfully Added! =====|
C 1 30
sended
Course changed!

D:\7 s\indiv\client\Debug\server.exe
Added Finance Data: ID: 1, Name: RUB Short name: RU, Course: 70.23, absCourse: 0
.00, conCourse 0.00

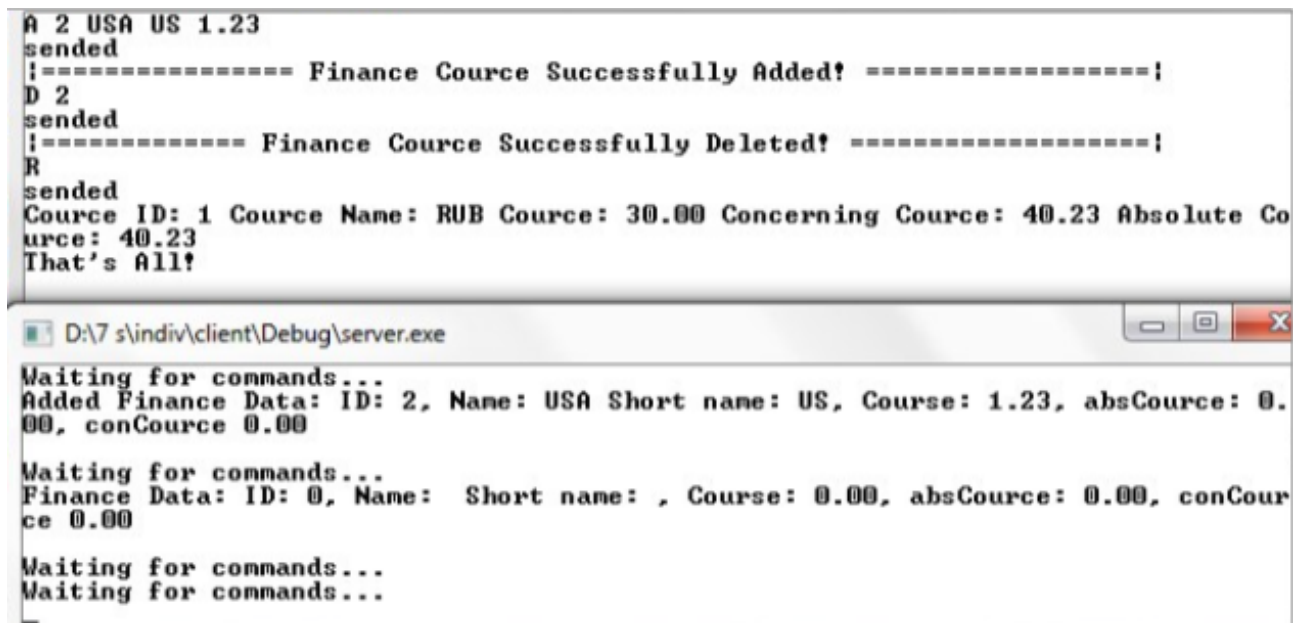
Waiting for commands...
Finance Data: ID: 1, Name: RUB Short name: RU, Course: 30.00, absCourse: 40.23,
conCourse 40.23

Waiting for commands...
```

Рис. 7: Изменение курса валюты

На сервере получили сообщение о том, что курс валюты изменился.
 $\text{absCourse} = 70.23 - 30 = 40.23$

Добавим еще одну валюту (A 2 USA US 1.23), удалим её (D 2) и прочитаем список всех валют (R)



```
A 2 USA US 1.23
sended
!===== Finance Course Successfully Added! =====!
D 2
sended
!===== Finance Course Successfully Deleted! =====!
R
sended
Course ID: 1 Course Name: RUB Course: 30.00 Concerning Course: 40.23 Absolute Co
urse: 40.23
That's All!

Waiting for commands...
Added Finance Data: ID: 2, Name: USA Short name: US, Course: 1.23, absCourse: 0.
00, conCourse 0.00

Waiting for commands...
Finance Data: ID: 0, Name:  Short name: , Course: 0.00, absCourse: 0.00, conCour
ce 0.00

Waiting for commands...
Waiting for commands...
```

Рис. 8: Удаление валюты

Вывести меню еще раз командой M

```
M
Main menu:

=====Commands:=====|=====Semantic:=====
Add New Finance Position:|A <ID> <NAME> <SHORT NAME> <COURSE>
Delete Finance Position: |D <ID>
Read All Finance Courcess:|R
Add Finance Courcess:    |C <ID> <COURSE>
Exit Programm:           |O
Show Menu:               |M
Show History              |H <ID>
=====
Enter your command... And PRAY!....
=====
```

Рис. 9: Вывод меню

Вывести историю изменения валюты командой H <ID>

```
H 1
sended
History: 0> 70.23
History: 1> 30.00
History: 2>
History: 3>
History: 4>
History: 5>
History: 6>
History: 7>
History: 8>
History: 9>
```

Рис. 10: Вывод истории

4.3 Обработка ошибок и предупреждений клиентского приложения

4.3.1 Неполный список аргументов

Исходные данные: A 1

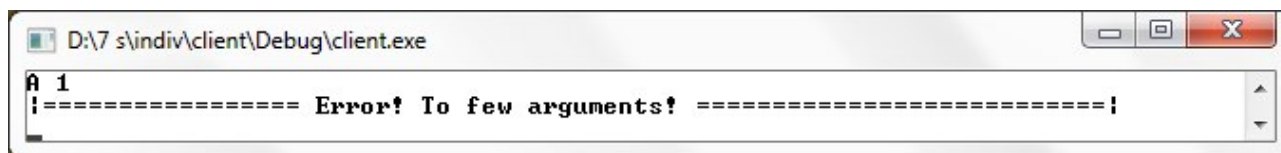


Рис. 11: Обработка неполного количества аргументов

4.3.2 Избыток цифр в ID

Исходные данные: A 231241214 RUB RU 12

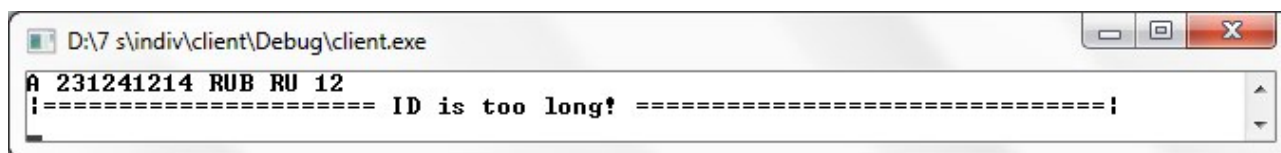


Рис. 12: Обработка избытка цифр в ID

4.3.3 Некорректный ID

Исходные данные: A 34 RUB RU 12

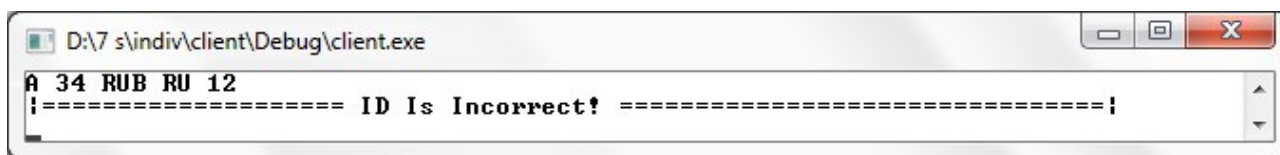


Рис. 13: Обработка некорректного ID

4.3.4 Слишком длинное название <NAME>

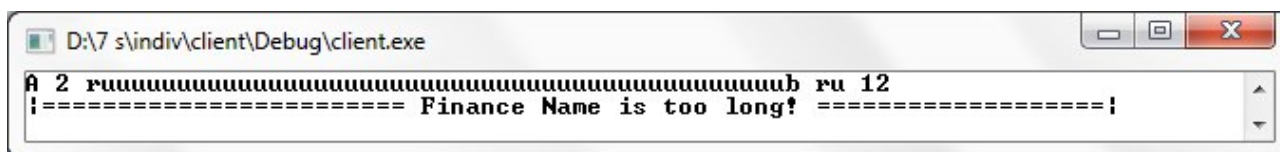
[illegible]

Рис. 14: Обработка <Name>

4.3.5 Слишком длинное <SHORT NAME>

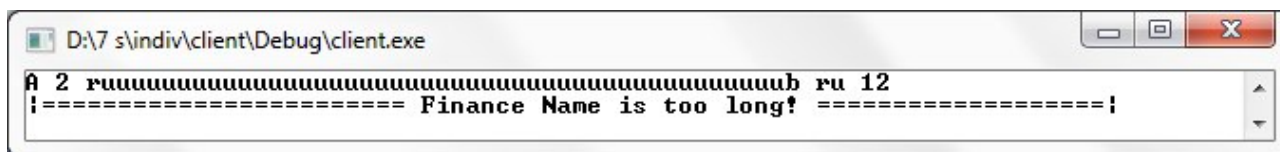
[illegible]

Рис. 15: Обработка <Short name>

4.3.6 Курс с таким ID не найден

Исходные данные: D 4

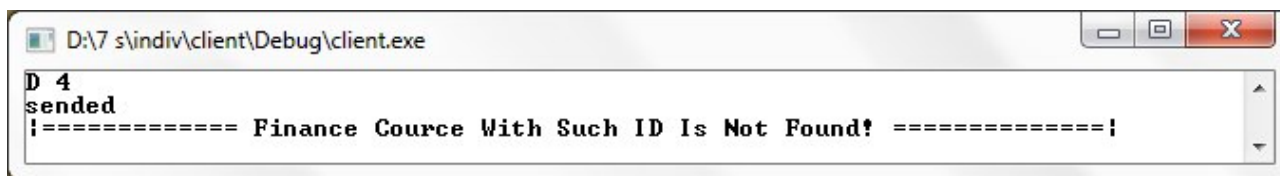


Рис. 16: Обработка <Short name>

5 Вывод

В данной лабораторной работе были реализованы клиент-серверные программы выставления курса валют, с написанием собственного протокола на основе TCP. Протокол был реализован на языке C++ для операционной системы Windows.

На примере данной разработки были изучены основные приемы использования протокола транспортного уровня ТСР – транспортного механизма, предоставляющего поток данных, с предварительной установкой соединения. Его преимуществом является достоверность получаемых данных (за счет осуществления повторного запроса данных в случае их потери, устранения дублирования при получении копий одного пакета).

Однако TCP может не подходить в некоторых ситуациях обмена по сети вследствие медленной (по сравнению с UDP) работы. Например, передавая по сети данные требующие быстрого отклика в реальном времени, необходимо соблюдать жесткие временные рамки, которые не может выдержать протокол TCP.

В ходе данной работы я так же ознакомился с протоколом UDP и реализовал клиент-серверное приложение платежной системы. По сравнению с TCP, UDP – более простой, основанный на сообщениях протокол без установления соединения, однако требует дополнительного контроля доставки сообщений ввиду следующих особенностей:

- Ненадежность – когда сообщение посылается, неизвестно, достигнет ли оно своего назначения – оно может потеряться по пути.
- Неупорядоченность – если два сообщения отправлены последовательно, порядок их получения не может быть предугадан.

Исходя из этого, UDP наиболее часто используется чувствительными ко времени приложениями, когда небольшие потери не играют большой роли. Если же нужна надежность, то предпочтительнее использовать TCP.