

Галкина Таисия Олеговна БПИ213
Вариант №27

Условие:

Задача про экзамен. Преподаватель проводит экзамен у группы студентов. Каждый студент получает свой билет, сообщает его номер и готовит письменный ответ. Подготовив ответ, он передает его преподавателю. Преподаватель просматривает ответ и сообщает студенту оценку. Студент, дождавшись результата, уходит с экзамена. Требуется создать приложение, моделирующее действия преподавателя и студентов, каждый из которых представлен отдельным процессом. Преподаватель — сервер. Каждый студент — отдельный клиент.

Оценка 4-5:

1. Запуск программы:

1.1.Сервер:

1.1.1. Подаются в командную строку *<IP-адрес> <порт сервера>*
<количество клиентов>

1.2.Клиенты

1.2.1. Подаются в командную строку *<IP-адрес сервера> <порт сервера>*

1.3.Сначала запускается сервер, а только потом клиенты

Взаимодействие между сущностями:

1. Сервер запускается на введенном IP-адресе и порту. Сервер создает сокет, который прослушивает на заданном порту и ожидает подключений клиентов
2. Когда клиент-студент запускает свою программу, он подключается к серверу, используя IP-адрес и порт сервера. Клиент создает сокет, устанавливает соединение с сервером и отправляет запрос на получение номера билета
3. Сервер принимает соединение от клиента и создает новый сокет для обмена данными с этим клиентом
4. Сервер генерирует случайный номер билета и отправляет его клиенту через созданный сокет
5. Клиент получает номер билета от сервера и отправляет свой «ответ» (просто посылает «Ответ на билет №?») на сервер через созданный сокет.
6. Сервер принимает ответ от клиента
7. Сервер генерирует случайную оценку и отправляет ее клиенту через созданный сокет.
8. Клиент получает оценку от сервера и завершает свое выполнение
9. Сервер закрывает соединение с клиентом и переходит к обслуживанию следующего клиента. Работа сервера прекращается тогда, когда

обработаны все клиенты (их количество мы получаем из командной строки)

10. После обслуживания всех клиентов сервер закрывает свой сокет и завершает свою работу

Таким образом выполнены пункты 1-8

Скриншоты по выполнению программ:

Работа сервера:

```
~/idzz3$ ./t 127.0.0.1 8080 3
Ожидание подключения клиентов...
Отправлен номер билета: 52
Студент отправил ответ: Ответ на билет 52
Отправлена оценка: 3
Отправлен номер билета: 83
Студент отправил ответ: Ответ на билет 83
Отправлена оценка: 6
Отправлен номер билета: 7
Студент отправил ответ: Ответ на билет 7
Отправлена оценка: 7
```

Работа клиентов:

```
~/idzz3$ ./s 127.0.0.1 8080
Получен номер билета: 52
Отправлен ответ: Ответ на билет 52
Получена оценка: 3
```

```
~/idzz3$ ./s 127.0.0.1 8080
Получен номер билета: 83
Отправлен ответ: Ответ на билет 83
Получена оценка: 6
```

```
~/idzz3$ ./s 127.0.0.1 8080
Получен номер билета: 7
Отправлен ответ: Ответ на билет 7
Получена оценка: 7
```

Оценка 6-7:

1. Запуск программы:

1.1. Сервер:

1.1.1. Подаются в командную строку *<IP-адрес> <порт сервера>*
<порт монитора> <количество клиентов>

1.2. Монитор

1.2.1. Подаются в командную строку *<IP-адрес> <порт сервера>*
<порт монитора>

1.3. Клиенты

1.3.1. Подаются в командную строку *<IP-адрес сервера> <порт сервера>*

1.4. Сначала запускается сервер, потом монитор, а только потом клиенты

Взаимодействие между сущностями (дополнение относительно прошлой оценки):

1. Создается сокет монитора, который также привязывается к IP-адресу сервера, но использует другой порт. Сокет монитора также прослушивает входящие подключения
2. Ожидается подключение монитора к серверу с помощью функции `assert()`. Когда монитор подключается, устанавливается соединение сокета монитора
3. В цикле `while`, клиент-монитор принимает данные от сервера с помощью функции `recv` и выводит их на экран. То есть весь главный процесс отображается в клиенте-монитор
4. Когда все клиенты обслужены, сервер отправляет сообщение "stop" на сокет монитора с помощью функции `send_to_monitor()` и закрывает все сокеты
5. После чего монитор тоже прекращает свою работу

Таким образом выполнены пункты 1-8

Скриншоты по выполнению программ:

Работа сервера:

```
~/idz3$ ./t 127.0.0.1 8082 8083 2
Ожидание подключения монитора...
Ожидание подключения клиентов...
```

Работа клиента-монитор:

```
~/idz3$ ./m 127.0.0.1 8082 8083
Студенту-0 отправлен номер билета: 1
Студент-0 отправил ответ: Ответ на билет 1
Преподаватель отправил оценку: 4 студенту-0
Студенту-1 отправлен номер билета: 1
Студент-1 отправил ответ: Ответ на билет 1
Преподаватель отправил оценку: 9 студенту-1
```

Работа клиентов-студентов:

```
~/idz3$ ./s 127.0.0.1 8082  
Получен номер билета: 1  
Отправлен ответ: Ответ на билет 1  
Получена оценка: 9
```

```
~/idz3$ ./s 127.0.0.1 8082  
Получен номер билета: 1  
Отправлен ответ: Ответ на билет 1  
Получена оценка: 4
```