

*Галкина Таисия Олеговна БПИИ213*  
*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> <порт>*

1.3.Сначала запускается сервер, а только потом клиенты

Взаимодействие между сущностями:

1. Взаимодействие между сервером и клиентом осуществляется с использованием протокола UDP
2. Сервер запускается на введенном IP-адресе и порту. Сервер создает сокет, который прослушивает на заданном порту и ожидает подключений клиентов
3. Когда клиент-студент запускает свою программу, он подключается к серверу, используя IP-адрес и порт сервера
4. Клиент отправляет серверу запрос "get\_ticket", чтобы получить номер билета и установить связь
5. Сервер принимает запрос от клиента, проверяет полученное сообщение и генерирует случайный номер билета. Затем сервер отправляет номер билета клиенту
6. После клиент отправляет «ответ» на номер билета обратно серверу
7. Сервер принимает «ответ» от клиента и генерирует и отправляет случайную оценку клиенту
8. Клиент принимает оценку от сервера с помощью функции «recvfrom» и выводит ее на экран
9. После завершает свою работу
- 10.После повторяются пункты 4-9 у сервера

11. В конце обслуживания (он знает общее количество клиентов) всех клиентов сервер закрывает свой сокет и завершает свою работу

Таким образом выполнены все пункты

Скриншоты по выполнению программ:

Работа сервера:

```
~/os4$ ./t 127.0.0.1 8080 3
Сервер начал работу
Отправлен билет номер 827 студенту 1
Получен ответ на номер 827 от студента 1
Отправлена оценка 6 студенту 1
Отправлен билет номер 615 студенту 2
Получен ответ на номер 615 от студента 2
Отправлена оценка 8 студенту 2
Отправлен билет номер 602 студенту 3
Получен ответ на номер 602 от студента 3
Отправлена оценка 9 студенту 3
```

Работа клиентов:

```
~/os4$ ./s 127.0.0.1 8080
Получен билет: 827
Отправлен ответ на билет номер 827
Получена оценка - 6
```

```
~/os4$ ./s 127.0.0.1 8080
Получен билет: 615
Отправлен ответ на билет номер 615
Получена оценка - 8
```

```
~/os4$ ./s 127.0.0.1 8080
Получен билет: 602
Отправлен ответ на билет номер 602
Получена оценка - 9
```

### Оценка 6-7:

1. Запуск программы:

1.1. Сервер:

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

1.2. Монитор

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

1.3. Клиенты

- 1.3.1. Подаются в командную строку *<IP-адрес> <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. Клиент-монитор отправляет серверу запрос "get\_messages", чтобы установить связь
3. Сервер ожидает сообщение от монитора с помощью функции «recvfrom»
4. В цикле while, клиент-монитор принимает данные от сервера с помощью функции «recvfrom» и выводит их на экран. То есть весь главный процесс отображается в клиенте-монитор
5. Когда все клиенты обслужены, сервер отправляет сообщение "stop" на сокет монитора и прекращает работу
6. После чего монитор тоже прекращает свою работу

Таким образом выполнены все пункты

Скриншоты по выполнению программ:

Работа сервера:

```
~/os4$ ./t 127.0.0.1 8080 8081 3
Сервер начал работу
```

Работа клиента-монитор:

```
~/os4$ ./m 127.0.0.1 8080
Отправлен билет номер 357 студенту 1
Получен ответ на номер 357 от студента 1
Отправлена оценка 8 студенту 1
Отправлен билет номер 760 студенту 2
Получен ответ на номер 760 от студента 2
Отправлена оценка 5 студенту 2
Отправлен билет номер 499 студенту 3
Получен ответ на номер 499 от студента 3
Отправлена оценка 8 студенту 3
```

Работа клиентов-студентов:

```
~/os4$ ./s 127.0.0.1 8080
Получен билет: 357
Отправлен ответ на билет номер 357
Получена оценка - 8
```

```
~/os4$ ./s 127.0.0.1 8080
```

Получен билет: 760

Отправлен ответ на билет номер 760

Получена оценка – 5

```
~/os4$ ./s 127.0.0.1 8080
```

Получен билет: 499

Отправлен ответ на билет номер 499

Получена оценка – 8