

27. **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.*

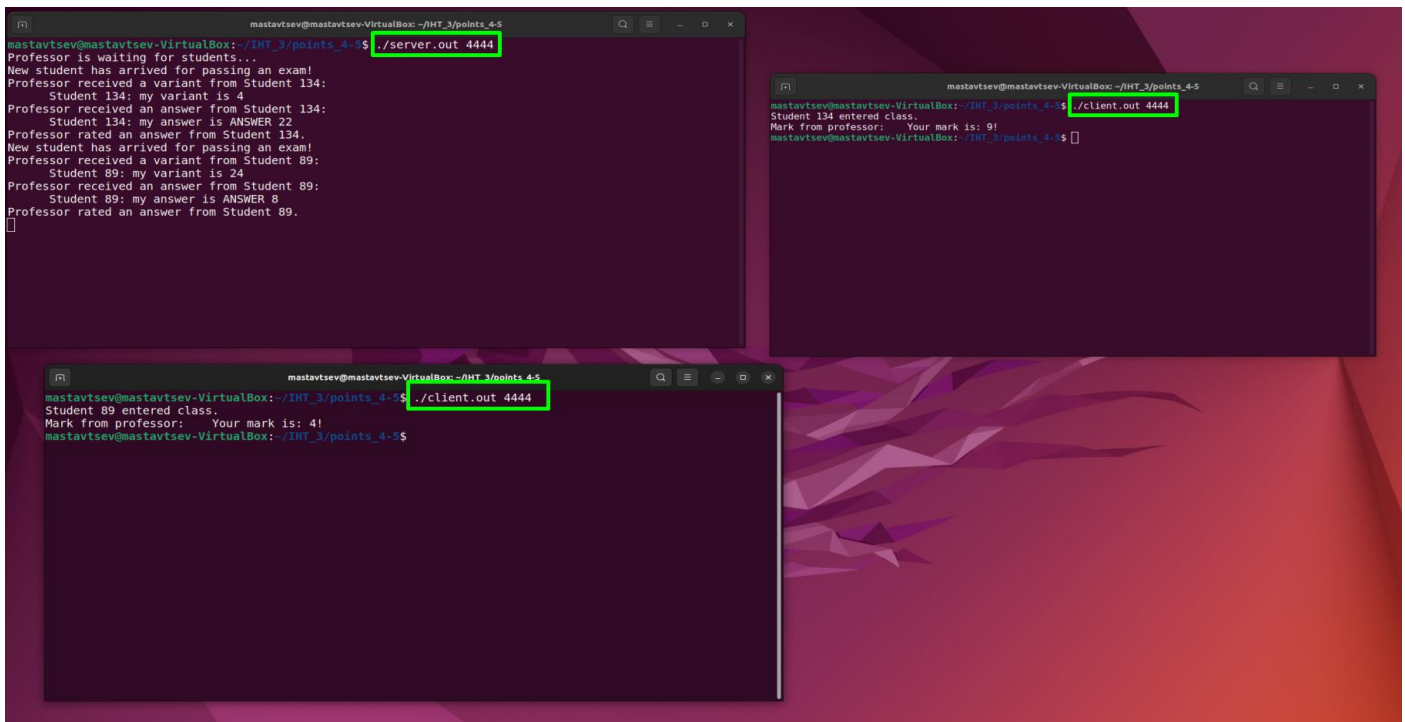
4 - 5 баллов

Файлы и каталоги:

server.c

client.c

Ввод и вывод данных (сервер и два клиента):



```
mastavtsev@mastavtsev-VirtualBox: ~/IHT_3/points_4-5
mastavtsev@mastavtsev-VirtualBox:~/IHT_3/points_4-5$ ./server.out 4444
Professor is waiting for students...
New student has arrived for passing an exam!
Professor received a variant from Student 134:
  Student 134: my variant is 4
Professor received an answer from Student 134:
  Student 134: my answer is ANSWER 22
Professor rated an answer from Student 134.
New student has arrived for passing an exam!
Professor received a variant from Student 89:
  Student 89: my variant is 24
Professor received an answer from Student 89:
  Student 89: my answer is ANSWER 8
Professor rated an answer from Student 89.

```

```
mastavtsev@mastavtsev-VirtualBox:~/IHT_3/points_4-5$ ./client.out 4444
Student 134 entered class.
Mark from professor: Your mark is: 9!
mastavtsev@mastavtsev-VirtualBox:~/IHT_3/points_4-5$

```

```
mastavtsev@mastavtsev-VirtualBox:~/IHT_3/points_4-5$ ./client.out 4444
Student 89 entered class.
Mark from professor: Your mark is: 4!
mastavtsev@mastavtsev-VirtualBox:~/IHT_3/points_4-5$

```

Параметрами для исполняемых файлов являются:

server.c – порт сервера, на котором он прослушивает клиентов (4444)

client.c – тот же номер порта, на который посылаются сообщения (4444)

Важно отметить, что здесь и далее параметрами не будет являться значение IP, так как в виртуальной среде Virtual Box при указании конкретного значения IP у меня вылетала ошибка Permission Denied, то есть система почему-то отказывала в доступе к localhost. Поэтому везде при конфигурации IP сокета указывался флаг INADDR_ANY.

Процесс сдачи экзамены (выходные данные):

Данный раздел актуален для всех программ, поэтому пояснения будут только тут.

Процесс сдачи экзамен происходит следующим образом:

1. Студент тянет билет -> сообщает номер варианта профессору (через канал/способ передачи информации, обозначенный в задании на конкретный балл).
2. Профессор получает сообщение от студента, выводит его и продолжает «принимать экзамен» - ожидает сообщения от других студентов.
3. После сообщения варианта студент «решает свой билет» и сообщает ответ преподавателю, так же необходимым в задаче способом (ответом является просто фраза ANSWER “random num”). Далее он ждёт, когда профессор сообщит ему оценку.
4. Профессор думает над оценкой студента (sleep(1)) и сообщает ему.
5. Студент, получив оценку от профессора, завершает работу.

Передача сообщений между процессами сервера (профессор) и клиентов (студенты) происходят за счёт протокола **UDP**.

К серверу могут подключаться одновременно сразу несколько клиентов. Ровно как преподаватель принимает экзамен сразу у группы студентов, сервер работает сразу с группой клиентов. Особенность протокола UDP заключается в том, что он способен сразу обрабатывать запросы от

множества клиентов. Поэтому в данном случае, в отличие от приложения с TCP протоколом, обработка клиентов (студентов) ведётся в рамках одного процесса.

Передача сообщений от студента к профессору происходит на основе структуры данных Message:

student_id - уникальный номер студента, используется для различия студентов

message_type - тип сообщения:

message_type == 0 - сообщение о номере варианта

message_type == 1 - ответ на вариант

message - само сообщение

6-7 баллов

Файлы и каталоги:

server.c

client.c

observer.c

Ввод и вывод данных (сервер, два клиента и единственный наблюдатель):

```
mastavtsev@mastavtsev-VirtualBox: ~/INT_3/points_6-7
mastavtsev@mastavtsev-VirtualBox:~/INT_3/points_6-7$ ./server.out 4444 4040
Professor is waiting for students...
Student 56 sent a variant number.
Professor received variant from Student 56
Student 56 sent an answer.
Professor received an answer from Student 56
Professor rated an answer from Student 56 with 9.
Student 19 sent a variant number.
Professor received variant from Student 19
Student 19 sent an answer.
Professor received an answer from Student 19
Professor rated an answer from Student 19 with 2.
█

mastavtsev@mastavtsev-VirtualBox:~/INT_3/points_6-7$ ./client.out 4444
Student 19 entered a class.
Mark from professor: Your mark is: 2!
mastavtsev@mastavtsev-VirtualBox:~/INT_3/points_6-7$ █

mastavtsev@mastavtsev-VirtualBox:~/INT_3/points_6-7$ ./observer.out 4040
Observer started to observe.
Student 56 sent a variant number.
Professor received variant from Student 56
Student 56 sent an answer.
Professor received an answer from Student 56
Professor rated an answer from Student 56 with 9.
Student 19 sent a variant number.
Professor received variant from Student 19
Student 19 sent an answer.
Professor received an answer from Student 19
Professor rated an answer from Student 19 with 2.
█

mastavtsev@mastavtsev-VirtualBox:~/INT_3/points_6-7$ ./client.out 4444
Student 56 entered a class.
Mark from professor: Your mark is: 9!
mastavtsev@mastavtsev-VirtualBox:~/INT_3/points_6-7$ █
```

Параметрами для исполняемых файлов являются:

server.c – порт сервера, на котором он прослушивает клиентов (4444) и порт сервера, на котором он устанавливает соединение с наблюдателем (4040)

client.c – номер порта, на который посылаются сообщения от сервера клиенту (4444)

observer.c - номер порта, на который посылаются сообщения от сервера наблюдателю (4040)

Важно отметить, что важен порядок запуска процессов различных агентов. Сначала необходимо запустить процесс сервера, далее запускается процесс наблюдателя, и уже после желаемое количество процессов клиентов.

Общая схема решаемой задачи:

Отличием данной задачи является наличие ещё одного типа клиента сервера – наблюдателя. Для корректного взаимодействия с двумя различными типами клиентов на сервера создаётся два различных сокета, которые прослушивают два различных порта. На одном порту происходит взаимодействие с множеством клиентов – студентами, на другом – отправка сообщений о состоянии системы наблюдателю.

8 баллов

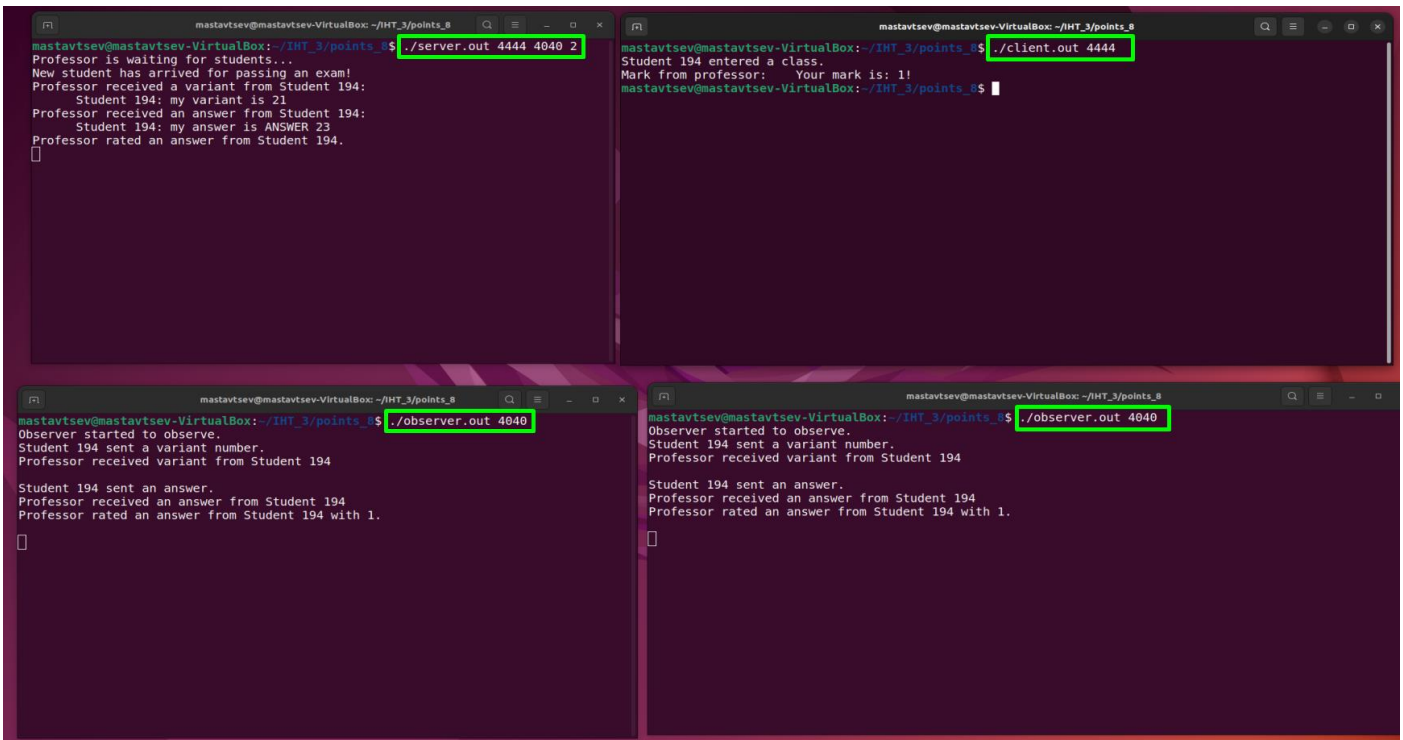
Файлы и каталоги:

server.c

client.c

observer.c

Ввод и вывод данных (сервер, один клиент и два наблюдателя):



The image displays four terminal windows arranged in a 2x2 grid, all running on a system named 'mastavtsev@mastavtsev-VirtualBox' with the directory '~/IHT_3/points_8'. Each window shows the output of a different program, with specific command-line arguments highlighted in green boxes.

- Top-left window:** Running `./server.out 4444 4040 2`. The output shows a professor waiting for students, a student arriving, and the professor receiving an answer from Student 194.
- Top-right window:** Running `./client.out 4444`. The output shows Student 194 entering a class and receiving a mark of 1! from the professor.
- Bottom-left window:** Running `./observer.out 4040`. The output shows the observer starting to observe, receiving a variant number from Student 194, and then receiving an answer from Student 194, which is rated with 1.
- Bottom-right window:** Running `./observer.out 4040`. The output shows the observer starting to observe, receiving a variant number from Student 194, and then receiving an answer from Student 194, which is rated with 1.

Параметрами для исполняемого файла являются:

server.c – (1) порт сервера, на котором он прослушивает клиентов (4444) и (2) порт сервера, на котором он устанавливает соединение с наблюдателем (4040) и (3) количество наблюдателей N

client.c – номер порта, на который посылаются сообщения от сервера клиенту (4444)

observer.c - номер порта, на который посылаются сообщения от сервера наблюдателю (4040)

Спецификой задачи на данный балл является необходимость осуществлять рассылку информации о взаимодействия сервера-профессора и клиента-студента сразу нескольким клиентам-наблюдателям.

Важно отметить, что важен порядок запуска процессов различных агентов. Сначала необходимо запустить процесс сервера, далее запускаются N процессов наблюдателей, и уже после желаемое количество процессов клиентов.

Благодаря тому, что в данном приложении нам необходимо использовать протокол UDP, мы воспользуемся методом широковещательной рассылки для распространения сообщений о состоянии системы между множеством клиентов – наблюдателей.

Данное приложение также использует два порта для взаимодействия с двумя различными клиентами. Ключевым отличием является создание сокета для взаимодействия клиентом наблюдателем. При создании сокета необходимо установить его в состояние широковещательной рассылки, а также при создании адресной структуры указать интерфейс широковещательной рассылки (`inet_addr(BROADCAST_ADDR)`).

Сергей Александрович, если Вы будете проверять эту работу, то хочу сказать Вам спасибо за этот курс и за курс по ABC. Я действительно считаю, что они были для меня полезны, и Вы с Александром Ивановичем проделали отличную работу. Спасибо!