

Федеральное государственное автономное образовательное
учреждение высшего образования
**"НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
"ВЫСШАЯ ШКОЛА ЭКОНОМИКИ"**

Факультет Компьютерных наук
Департамент программной инженерии

Пояснительная записка

к домашнему заданию по дисциплине
"Архитектура вычислительных систем"

Симоновича Иван Сергеевича

БПИ 193-2 19 Вариант

Москва 2020

Формулировка задания:

Вариант 19. У одной очень привлекательной студентки есть N поклонников. Традиционно в день св. Валентина очень привлекательная студентка проводит романтический вечер с одним из поклонников. Счастливый избранник заранее не известен. С утра очень привлекательная студентка получает N «валентинок» с различными вариантами романтического вечера. Выбрав наиболее заманчивое предложение, студентка извещает счастливчика о своем согласии, а остальных – об отказе. Требуется создать многопоточное приложение, моделирующее поведение студентки. При решении использовать парадигму «клиент-сервер» с активным ожиданием.

Разработать программу с применением OpenMP и протестировать ее на нескольких примерах.

Решение задания:

При решении использовалась парадигма «клиент-сервер» с активным ожиданием. Согласно этой парадигме несколько клиентов и один сервер совместно с промежуточным программным обеспечением и средой взаимодействия образуют единую систему, обеспечивающую распределенные вычисления, анализ и представление данных. Если точнее, то клиентский поток запрашивает сервер и после ожидает ответа, в то время как сервер ждет запроса от клиента и после этого выполняет свою логику в зависимости от запроса и отправляет ответ на запрос клиенту. Иногда клиент может «исполнить» роль сервера, если сам будет получать запросы. Аналогично сервер будет выступать в роли клиента, если ему потребуется обращаться с запросами к другим программам.

Источники, из которых брал информацию:

<https://pro-prof.com/forums/topic/parallel-programming-paradigms> - про парадигму клиент-сервер

<https://bagdar.info/modele-klient-server-razdelenie-prilojenij-po-urovnyam-tipi-kl.html> - про парадигму клиент-сервер

<http://ccfit.nsu.ru/arom/data/openmp.pdf> - информация по openmp

<https://www.cyberforum.ru/blogs/18334/blog2965.html> - информация по openmp

При запуске в программу вводится положительное число n большее нуля - количество клиентов сервера. Верхнее ограничение n равно 250, оно нужно для корректной работы программы, так как при слишком большом количестве консоль засоряется, и избегания чрезмерного использования ресурсов процессора. В программе присутствует проверка и повтор ввода, при неверных значениях.

Стоит отметить, что программа разрабатывалась в среде CLion, так как в Visual Studio 2019 не подключались нужные для разработки библиотеки, а именно `#include <unistd.h>` и `#include <omp.h>`. После внесения нужных изменений для функционирования библиотеки `omp.h` в `CMakeLists.txt` данный проект на CLion успешно заработал.

Для корректной реализации парадигмы клиент-сервер в программе были реализованы два класса `Valentine`, где хранятся данные об клиентах, и `Server`, где разработана логика сервера.

После успешного ввода в `main` создается массив из 8 фраз и указатель на сервер, куда передается информация о количестве клиентов.

После этого запускается $n+1$ параллельных потоков `#pragma omp parallel num_threads(n+1)`, где 0 поток будет потоком сервера, а остальные n потоков будут потоками клиентов. Ниже представлен фрагмент кода с запуском потоков и с проверкой на номер потока `omp_get_thread_num()`.

```
#pragma omp parallel num_threads(n+1) //Выделяем n+1 поток n для клиентов-
поколонников и 1 для сервера-студентки
{
    if (omp_get_thread_num() == 0) {
        printf("Сервер запускается...\n");
        server->serverMain(); //Запускаем метод сервера
        printf("Сервер засыпает, спят усталые игрушки, клиенты спят\n");
    } else {
        //Запускаем метод в потоке каждого из n клиентов от (1 до n+1)
        clientMain(omp_get_thread_num(), phrases[(rand() +
omp_get_thread_num()) % 8], server);
    }
}
```

При запуске потока для сервера `if (omp_get_thread_num() == 0)` вызовется метод `serverMain` экземпляра `server`, где запустится логика ожидания сообщений от каждого потока и логика выбора победителя после ожидания. Сервер заканчивает работу, когда каждый клиент после выбора победителя получит ответ о своей победе или поражении. Соответственно в этом методе были написаны два ожидания `while`, первое для проверки, что сообщения от всех клиентов поступили, а второе, что все клиенты получили ответ. Ниже представлен данный метод с подробными комментариями.

```

/// <summary>
/// Основная логика работы сервера, запускается в потоке
/// </summary>
void serverMain() {
    srand(time(NULL));
    printf("[SERVER] Начался новый день. Жду, пока утро закончится.\n");

    //Задержка перед определением победителя на сервере клиента, нужна чтобы
    все клиенты отправили данные
    while (size > count) {
        sleep(1);
    }

    printf("[SERVER] Утро закончилось. Получено %d писем.\n", (int)
letters.size());
    winner = rand() % letters.size(); //Определяем индекс победителя, после
этого isResponseReady вернет true
    //Указатель на письмо победителей
    Valentine *winnerLetter = letters[winner];

    //Вывод инфы о победителе
    printf("[SERVER] Победило письмо номер %d из комнаты %d с текстом
\"%s\".\n", winner + 1,
        winnerLetter->getNum(), winnerLetter->getText().c_str());

    //Проверка, что все клиенты получили информацию, после чего сервер
    прекращает работу
    while (count > 0)
        sleep(1);
}

```

В это же время для каждого клиента в его потоке вызывается метод ClientMain, куда передаются номер клиента `omp_get_thread_num()`, случайная фраза из массива `phrases[8]` и сервер `server`, на который он должен отправлять запросы. В этом методе клиент сообщает что начал писать письмо, после чего создается объект `Valentine` содержащий информацию о тексте письма и его номере, после чего поток засыпает на случайное время, эмулируя реального клиента и его трату времени на написание этого письма. Сам `srand()` задается в начале метода `clientMain`, его сид зависит от номера каждого клиента и размера его введенного текста.

```

/// <summary>
/// Логика работы каждого клиента, запускается для каждого клиента в
отдельном потоке
/// </summary>
/// <param name="roomNumber"></param>
/// <param name="text"></param>
/// <param name="server"></param>
void clientMain(int roomNumber, string text, Server *server) {
    //Обновляем рандом, чтобы для каждого потока был свой
    srand(roomNumber + text.length());
    printf("[CLIENT №%d] Начинаю писать письмо...\n", roomNumber);
    //Создаем письмо клиента
    Valentine letter(text, roomNumber);
}

```

```

//Тратит псевдослучайное время на написание письма
sleep(rand() % 3);

int letterNumber = server->enqueueRequest(&letter); //Клиент отправляет
валентинку на сервер
printf("[CLIENT №%d] Отправил письмо %d \"%s\", жду ответа!\n",
roomNumber, letterNumber, text.c_str()); //Выводит информацию об этом запросе

//Активное ожидание, пока сервер не определил победителя поток будет
ожидать
while (!server->isResponseReady())
    sleep(1);

//Получаем ответ с сервера по нашей валентинке, является ли клиент
победителем
bool response = server->getResponse(&letter);
//Выводим ответ в клиенте
if (response)
    printf("[CLIENT №%d] \tПолучил ответ.\tСчастливчик, сегодня тебя
выбрали!\n", roomNumber);
else
    printf("[CLIENT №%d] \tПолучил ответ.\tНет, студентка с тобой не
идет!\n", roomNumber);
}

```

В моем коде присутствуют необходимые для понимания работы комментарии, но я все же опишу еще некоторые вещи из работы метода `clientMain`. Если поток клиента уже поспал необходимое время, то он отправляет свой запрос с ссылкой на письмо на сервер

`int letterNumber = server->enqueueRequest(&letter);` Этот метод заносит письмо в вектор писем сервера и увеличивает счетчик полученных писем, а возвращает клиенту его порядковый номер письма из всех поступивших. Клиент выводит информацию об этом, а дальше происходит **активное ожидание** потока, то есть пока сервер не определит победителя все запущенные потоки будут ожидать.

```

while (!server->isResponseReady())
    sleep(1);

```

После того как сервер выбрал победителя, то есть когда метод `isResponseReady()` вернул `true`, клиент принимает ответ на свой запрос с сервера с помощью метода `server->getResponse(&letter)`, который возвращает `true`, если клиент является победителем. Затем клиент выводит информацию об ответе в консоль. Сам сервер перед тем, как клиенты начнут выводить информацию о своём ответе, после получения всех писем пишет их количество, а также письмо и номер комнаты победителя (номер самого клиента).

После того как сервер ответил всем клиентам он заканчивает работу и программа завершается ожиданием ответа от её пользователя `system("pause")`. Будет выведено:

“Для продолжения нажмите любую клавишу . . .”

Клиенты могут отправлять запросы на сервер в случайном порядке, так как у каждого из них перед этим случайное время задержки (сна), после которого они уже и отправляют свой запрос (см. примеры тестов).

Все подробные комментарии для каждого метода содержатся в коде программы, выше я попытался их ужать в более емкое описание без упоминания незначительных деталей.

Вся информация о действиях клиентов и сервера выводится в консоль. Если вывод происходит на сервере, то в начале предложения будет выведено [SERVER], если вывод происходит в клиенте, то в начале выведется [CLIENT *i*], где *i* – номер клиента. Также при отправке письма в консоль выводится его порядковый номер при поступлении на сервер, например “...Отправил письмо 14...” означает, что письмо поступило 14-ым на сервер. После конца ожидания сервером запросов от всех клиентов (поклонников) выводится информация о поступивших от них валентинках и о выбранном студенткой (сервером) победителе.

Тестирование программы:

Случаи некорректного ввода:

Несколько возможных случаев некорректного ввода разобраны на одном скриншоте. При вводе неподходящей строки или числа программа запросит повтор ввода и продолжит выполняться после корректного ввода:

```
"D:\CLion Programs\cmake-build-debug\CLion_Programs.exe"
Оформите ввод количества валентинок:
251
Введено отрицательное число, ноль или число большее чем 250
Повторите ввод
0
Введено отрицательное число, ноль или число большее чем 250
Повторите ввод
-1
Введено отрицательное число, ноль или число большее чем 250
Повторите ввод
2.5
Ошибка ввода!
Повторите ввод
Дайте любви рому
Ошибка ввода!
Повторите ввод
A
Ошибка ввода!
Повторите ввод
!
Сервер запускается...
```

Примеры корректного ввода:

Ниже приведены примеры тестов для $n > 0$

Для $n=1$

```
"D:\CLion Programs\cmake-build-debug\CLion_Programs.exe"
Оформите ввод количества валентинок:
1
Сервер запускается...
[SERVER] Начался новый день. Жду, пока утро закончится.
[CLIENT №1] Начинаю писать письмо...
[CLIENT №1] Отправил письмо 1 "Да, любил я сладко покодить, но больше всего я люблю тебя", жду ответа!
[SERVER] Утро закончилось. Получено 1 писем.
[SERVER] Победило письмо номер 1 из комнаты 1 с текстом "Да, любил я сладко покодить, но больше всего я люблю тебя".
Сервер засыпает, спят усталые игрушки, клиенты спят
[CLIENT №1] Получил ответ. Счастливчик, сегодня тебя выбрали!
Для продолжения нажмите любую клавишу . . .
```

Для n=5. Видно, что порядок отправки и выбора победителя каждый раз новый.

```
"D:\CLion Programs\cmake-build-debug\CLion_Programs.exe"
Оформите ввод количества валентинок:
5
Сервер запускается...
[SERVER] Начался новый день. Жду, пока утро закончится.
[CLIENT №2] Начинаю писать письмо...
[CLIENT №3] Начинаю писать письмо...
[CLIENT №5] Начинаю писать письмо...
[CLIENT №4] Начинаю писать письмо...
[CLIENT №1] Начинаю писать письмо...
[CLIENT №3] Отправил письмо 1 "Дед мороз тебе свою любовь принес", жду ответа!
[CLIENT №1] Отправил письмо 2 "Да, любил я сладко покодить, но больше всего я люблю тебя", жду ответа!
[CLIENT №5] Отправил письмо 3 "Давай выходи", жду ответа!
[CLIENT №4] Отправил письмо 4 "Я никогда не объявлю тебе забастовку", жду ответа!
[CLIENT №2] Отправил письмо 5 "Любимка", жду ответа!
[SERVER] Утро закончилось. Получено 5 писем.
[SERVER] Победило письмо номер 2 из комнаты 1 с текстом "Да, любил я сладко покодить, но больше всего я люблю тебя".
[CLIENT №2] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №5] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №4] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №3] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №1] Получил ответ. Счастливчик, сегодня тебя выбрали!
Сервер засыпает, спят усталые игрушки, клиенты спят
Для продолжения нажмите любую клавишу . . .
```

```
"D:\CLion Programs\cmake-build-debug\CLion_Programs.exe"
Оформите ввод количества валентинок:
5
Сервер запускается...
[SERVER] Начался новый день. Жду, пока утро закончится.
[CLIENT №4] Начинаю писать письмо...
[CLIENT №2] Начинаю писать письмо...
[CLIENT №5] Начинаю писать письмо...
[CLIENT №3] Начинаю писать письмо...
[CLIENT №1] Начинаю писать письмо...
[CLIENT №3] Отправил письмо 1 "Дед мороз тебе свою любовь принес", жду ответа!
[CLIENT №1] Отправил письмо 2 "Да, любил я сладко покодить, но больше всего я люблю тебя", жду ответа!
[CLIENT №4] Отправил письмо 3 "Я никогда не объявлю тебе забастовку", жду ответа!
[CLIENT №5] Отправил письмо 4 "Давай выходи", жду ответа!
[CLIENT №2] Отправил письмо 5 "Любимка", жду ответа!
[SERVER] Утро закончилось. Получено 5 писем.
[SERVER] Победило письмо номер 5 из комнаты 2 с текстом "Любимка".
[CLIENT №2] Получил ответ. Счастливчик, сегодня тебя выбрали!
[CLIENT №4] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №5] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №3] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №1] Получил ответ. Нет, студентка с тобой не идет!
Сервер засыпает, спят усталые игрушки, клиенты спят
Для продолжения нажмите любую клавишу . . .
```

Для n=8. Аналогично видно, что при разных запусках выводы значительно отличаются (порядок, победитель и его номер комнаты)


```

Оформите ввод количества валентинок:
■
Сервер запускается...
[SERVER] Начался новый день. Жду, пока утро закончится.
[CLIENT №8] Начинаю писать письмо...
[CLIENT №3] Начинаю писать письмо...
[CLIENT №6] Начинаю писать письмо...
[CLIENT №7] Начинаю писать письмо...
[CLIENT №4] Начинаю писать письмо...
[CLIENT №2] Начинаю писать письмо...
[CLIENT №5] Начинаю писать письмо...
[CLIENT №1] Начинаю писать письмо...
[CLIENT №3] Отправил письмо 1 "Дед мороз тебе свою любовь принес", жду ответа!
[CLIENT №1] Отправил письмо 2 "Да, любил я сладко покодить, но больше всего я люблю тебя", жду ответа!
[CLIENT №8] Отправил письмо 3 "Сильнее пуджа я люблю только тебя", жду ответа!
[CLIENT №7] Отправил письмо 4 "Моему любимому хейтеру", жду ответа!
[CLIENT №4] Отправил письмо 5 "Я никогда не объявлю тебе забастовку", жду ответа!
[CLIENT №5] Отправил письмо 6 "Давай выходи", жду ответа!
[CLIENT №6] Отправил письмо 7 "Я твоё чудовище", жду ответа!
[CLIENT №2] Отправил письмо 8 "Любимка", жду ответа!
[SERVER] Утро закончилось. Получено 8 писем.
[SERVER] Победило письмо номер 5 из комнаты 4 с текстом "Я никогда не объявлю тебе забастовку".
[CLIENT №8]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №6]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №7]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №2]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №4]    Получил ответ.  Счастливчик, сегодня тебя выбрали!
[CLIENT №3]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №5]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №1]    Получил ответ.  Нет, студентка с тобой не идет!
Сервер засыпает, спят усталые игрушки, клиенты спят
Для продолжения нажмите любую клавишу . . .|

```

```

Оформите ввод количества валентинок:
■
Сервер запускается...
[SERVER] Начался новый день. Жду, пока утро закончится.
[CLIENT №5] Начинаю писать письмо...
[CLIENT №3] Начинаю писать письмо...
[CLIENT №3] Отправил письмо 1 "Дед мороз тебе свою любовь принес", жду ответа!
[CLIENT №6] Начинаю писать письмо...
[CLIENT №4] Начинаю писать письмо...
[CLIENT №7] Начинаю писать письмо...
[CLIENT №8] Начинаю писать письмо...
[CLIENT №2] Начинаю писать письмо...
[CLIENT №1] Начинаю писать письмо...
[CLIENT №1] Отправил письмо 2 "Да, любил я сладко покодить, но больше всего я люблю тебя", жду ответа!
[CLIENT №5] Отправил письмо 3 "Давай выходи", жду ответа!
[CLIENT №4] Отправил письмо 4 "Я никогда не объявлю тебе забастовку", жду ответа!
[CLIENT №7] Отправил письмо 5 "Моему любимому хейтеру", жду ответа!
[CLIENT №8] Отправил письмо 6 "Сильнее пуджа я люблю только тебя", жду ответа!
[CLIENT №6] Отправил письмо 7 "Я твоё чудовище", жду ответа!
[CLIENT №2] Отправил письмо 8 "Любимка", жду ответа!
[SERVER] Утро закончилось. Получено 8 писем.
[SERVER] Победило письмо номер 3 из комнаты 5 с текстом "Давай выходи".
[CLIENT №5]    Получил ответ.  Счастливчик, сегодня тебя выбрали!
[CLIENT №3]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №6]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №4]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №2]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №7]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №1]    Получил ответ.  Нет, студентка с тобой не идет!
[CLIENT №8]    Получил ответ.  Нет, студентка с тобой не идет!
Сервер засыпает, спят усталые игрушки, клиенты спят
Для продолжения нажмите любую клавишу . . .

```

Для n=10. Аналогично

```
CLion Programs
"D:\CLion Programs\cmake-build-debug\CLion_Programs.exe"
Оформите ввод количества валентинок:
10
Сервер запускается...
[SERVER] Начался новый день. Жду, пока утро закончится.
[CLIENT №3] Начинаю писать письмо...
[CLIENT №5] Начинаю писать письмо...
[CLIENT №6] Начинаю писать письмо...
[CLIENT №4] Начинаю писать письмо...
[CLIENT №2] Начинаю писать письмо...
[CLIENT №1] Начинаю писать письмо...
[CLIENT №8] Начинаю писать письмо...
[CLIENT №9] Начинаю писать письмо...
[CLIENT №10] Начинаю писать письмо...
[CLIENT №7] Начинаю писать письмо...
[CLIENT №3] Отправил письмо 1 "Дед мороз тебе свою любовь принес", жду ответа!
[CLIENT №1] Отправил письмо 2 "Да, любил я сладко покодить, но больше всего я люблю тебя", жду ответа!
[CLIENT №5] Отправил письмо 3 "Давай выходи", жду ответа!
[CLIENT №4] Отправил письмо 4 "Я никогда не объявляю тебе забастовку", жду ответа!
[CLIENT №8] Отправил письмо 5 "Сильнее пуджа я люблю только тебя", жду ответа!
[CLIENT №10] Отправил письмо 6 "Любимка", жду ответа!
[CLIENT №7] Отправил письмо 7 "Моему любимому хейтеру", жду ответа!
[CLIENT №6] Отправил письмо 8 "Я твоё чудовище", жду ответа!
[CLIENT №2] Отправил письмо 9 "Любимка", жду ответа!
[CLIENT №9] Отправил письмо 10 "Да, любил я сладко покодить, но больше всего я люблю тебя", жду ответа!
[SERVER] Утро закончилось. Получено 10 писем.
[SERVER] Победило письмо номер 1 из комнаты 3 с текстом "Дед мороз тебе свою любовь принес".
[CLIENT №5] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №6] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №4] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №2] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №8] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №10] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №9] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №3] Получил ответ. Счастливчик, сегодня тебя выбрали!
[CLIENT №7] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №1] Получил ответ. Нет, студентка с тобой не идет!
Сервер засыпает, спят усталые игрушки, клиенты спят
Для продолжения нажмите любую клавишу . . .
```

```
"D:\CLion Programs\cmake-build-debug\CLion_Programs.exe"
Оформите ввод количества валентинок:
10
[CLIENT №4] Начинаю писать письмо...
[CLIENT №1] Начинаю писать письмо...
[CLIENT №7] Начинаю писать письмо...
[CLIENT №5] Начинаю писать письмо...
[CLIENT №2] Начинаю писать письмо...
[CLIENT №3] Начинаю писать письмо...
[CLIENT №3] Отправил письмо 2 "Дед мороз тебе свою любовь принес", жду ответа!
[CLIENT №9] Начинаю писать письмо...
[CLIENT №8] Начинаю писать письмо...
Сервер запускается...
[SERVER] Начался новый день. Жду, пока утро закончится.
[CLIENT №1] Отправил письмо 1 "Да, любил я сладко покодить, но больше всего я люблю тебя", жду ответа!
[CLIENT №6] Начинаю писать письмо...
[CLIENT №10] Начинаю писать письмо...
[CLIENT №4] Отправил письмо 3 "Я никогда не объявлю тебе забастовку", жду ответа!
[CLIENT №7] Отправил письмо 4 "Моему любимому хейтеру", жду ответа!
[CLIENT №5] Отправил письмо 5 "Давай выходи", жду ответа!
[CLIENT №8] Отправил письмо 6 "Сильнее пуджа я люблю только тебя", жду ответа!
[CLIENT №10] Отправил письмо 7 "Любимка", жду ответа!
[CLIENT №2] Отправил письмо 8 "Любимка", жду ответа!
[CLIENT №9] Отправил письмо 9 "Да, любил я сладко покодить, но больше всего я люблю тебя", жду ответа!
[CLIENT №6] Отправил письмо 10 "Я твоё чудовище", жду ответа!
[SERVER] Утро закончилось. Получено 10 писем.
[SERVER] Победило письмо номер 7 из комнаты 10 с текстом "Любимка".
[CLIENT №1] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №6] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №10] Получил ответ. Счастливчик, сегодня тебя выбрали!
[CLIENT №4] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №7] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №2] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №5] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №3] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №9] Получил ответ. Нет, студентка с тобой не идет!
[CLIENT №8] Получил ответ. Нет, студентка с тобой не идет!
Сервер засыпает, спят усталые игрушки, клиенты спят
Для продолжения нажмите любую клавишу . . .
```