Дмитриев Дмитрий Игоревич, БПИ227

**Вариант 17. Условие задачи:** Задача о капиталистической гостинице. В гостинице 10 номеров с ценой 2000 УЕ, 10 номеров с ценой 4000 УЕ и 5 номеров с ценой 6000 УЕ. Клиент, зашедший в гостиницу, обладает некоторой (случайной) суммой и получает номер по своим финансовым возможностям, если тот свободен. При наличии денег и разных по стоимости номеров он выбирает случайный номер. Если доступных по цене свободных номеров нет, клиент уходит искать ночлег в другое место. Клиенты порождаются динамически и уничтожаются при освобождении номера или уходе из гостиницы при невозможности оплаты. Создать многопоточное приложение, моделирующее работу гостиницы. Каждого клиента и гостиницу (точнее ее администратора) моделировать отдельным потоком.

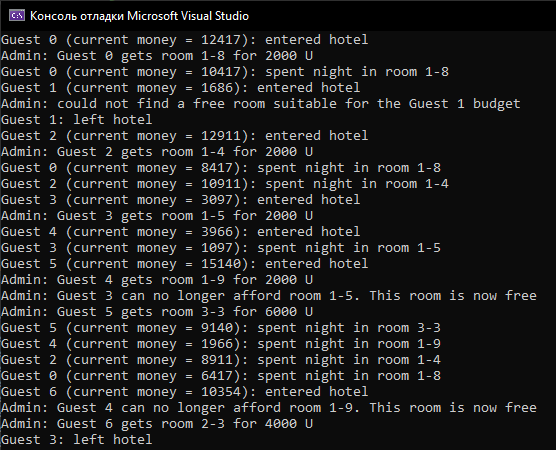
**Претендую на 8 баллов за выполнение задания**

(Для запуска программы нужно использовать **AVS-IDZ-4\x64\Debug\AVS-IDZ-4.exe** или **AVS-IDZ-4\AVS-IDZ-4\AVS-IDZ-4.cpp**)

**Прогрессирующий отчёт на баллы от 4 до 8:**

**4-5 баллов:**

1. Решил задачу 17 на C++, используя POSIX Thread. Отчёт составлен.
2. **Сценарий:** Есть отель и его администратор. В отеле 3 этажа, на 1 этаже 10 номеров по 2000 УЕ за ночь, на 2-ом – 10 номеров по 4000 УЕ и на 3-ем – 5 номеров по 6000 УЕ (нумерация номеров на этаже начинается с 1). В отель всё время приходят гости, админ должен дать им любой свободный номер, подходящий их бюджету, тогда этот номер занят, пока гость не съедет оттуда. Однако в случае нехватки денег или отсутствия подходящих свободных номеров админ должен отказать гостю в услугах. Если гостю отказали, он уходит, иначе гость каждый день обращается к админу, чтобы оплатить номер, если денег хватает на оплату, админ их снимает, если нет, гость уходит из отеля и номер становится свободным.
3. Модель параллельных вычислений моей программы напоминает модель (Много клиентов – 1 сервер), т.е. основной поток создаёт потоки гостей, а те отправляют запросы в общий буфер и засыпают, потом просыпается поток админа, обрабатывает запросы и засыпает, далее старые потоки гостей просыпаются и снова отправляют запросы (всё зацикливая), либо завершаются, а также параллельно продолжают появляться новые потоки гостей.
4. Входные данные не предусмотрены в задаче, т.к. всё рандомно генерится.
5. Реализовал консольное приложение, используя следующие синхропримитивы пакета POSIX: условные переменные и мьютексы.
6. Ввод задачей не предусмотрен.
7. Я рандомно генерировал суммы денег для гостей в диапазоне от 0 до 20000 УЕ, а также генерировал числа, чтобы случайно выбрать подходящий номер для гостя (диапазон от 0 до длины списка подходящих номеров не включительно).
8. Вывод должен быть интуитивно понятен, единственное поясню, что когда я пишу полный номер комнаты x-x, то слева от дефиса – этаж, а справа – номер комнаты на этаже.
9. Комментарии кода есть.
10. Пример работы программы:

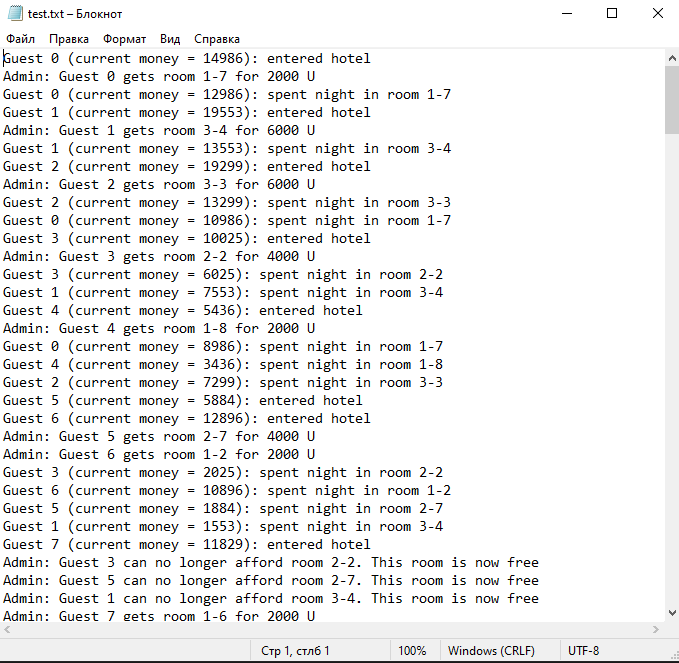


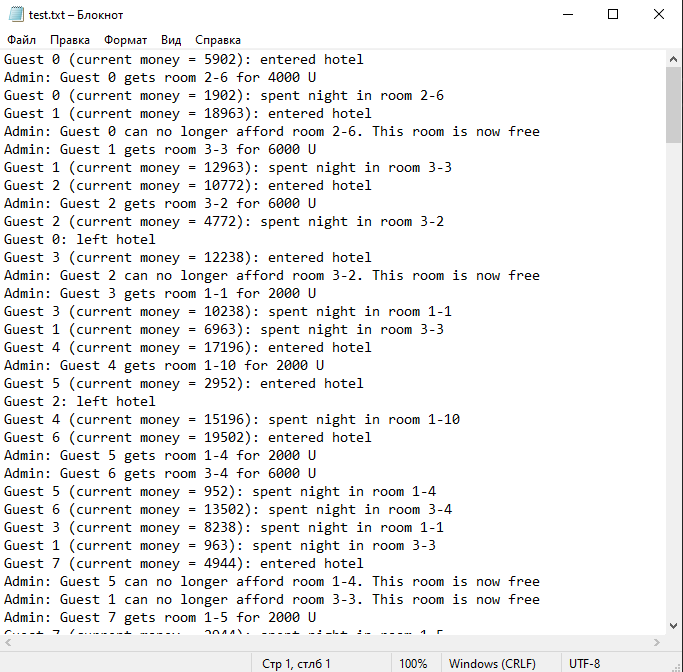
**6-7 баллов:**

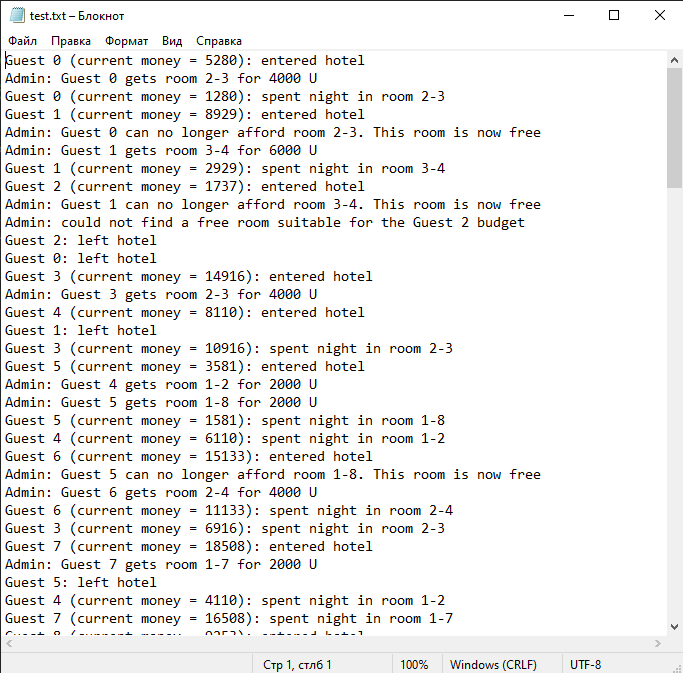
1. В основном потоке мы создаём потоки гостей (рандомно генерируя им бюджет), которые совершают запросы, хранящие информацию о госте (его id, деньги, есть ли у него комната уже, номер комнаты). Запросы хранятся в общем буфере (массиве), доступном для гостей и админа. Потом потоки гостей будят поток админа (сами гости засыпают, ожидая обработки запросов) и сами засыпают, поток админа обрабатывает эти запросы (прикрепляет подходящий номер к гостю, если такой есть; снимает деньги с гостя и выселяет его, если у него не хватает денег), будит гостей и засыпает, пока не появятся новые запросы и его не разбудят гости. Далее гости обрабатывают новую инфу о себе и, если нет мест или денег, они уходят из отеля (поток уничтожается), иначе заново отправляют запрос и цикл повторяется, также параллельно продолжают появляться новые потоки гостей.
2. Реализованы генерации рандомных чисел, упомянутые ранее.
3. Реализован ввод исходных данных из командной строки при запуске программы.

**8 баллов:**

1. В программу, наряду с выводом в консоль, добавлен вывод результатов в файл. Имя файла для вывода данных задается в командной строке как ее параметр.
2. Результаты работы программы выводятся на экран и записываются в файл, указанный пользователем.
3. Входные данные задачей не предусмотрены, кроме имени файла, куда заносятся результаты. Но бессмысленно делать конфиг файл для одного аргумента, когда в любом случае придётся вводить хотя бы один аргумент – имя самого конфига.
4. Входные данные задачей не предусмотрены, но приведу на этот раз 3 примера вывода в файл:







1. Т.к. ввод данных задачей не предусмотрен, я ввожу только путь файл, где буду хранить результаты программы, в командную строку.