

Соломонов Данила, БПИ203

Вариант 26.

Необходимо было разработать консольное приложение с использованием библиотеки **POSIX Threads** языка программирования **C** или **стандартной библиотеки** языка программирования **C++**. Разработка осуществлялась на языке программирования **C++**. При решении использовалась парадигма «клиент-сервер».

«Клиент — сервер» — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.

Подробнее: https://en.wikipedia.org/wiki/Client%E2%80%93server_model

26. 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. Требуется создать многопоточное приложение, моделирующее действия преподавателя и студентов. При решении использовать парадигму «клиент-сервер».

При решении задачи был использован объектно-ориентированный подход и введены следующие сущности: учитель (поток сервера) и студент (поток клиента). Учитель в коде представлен единственным потоком, который работает до тех пор, пока каждый из студентов не ответит на свой билет. К серверу поступают запросы от переменного количества студентов. Запросы бывают двух видов: на получение билета и на проверку ответа студента. В один момент времени учитель обрабатывает только один запрос от какого-либо студента, т. е. выдает или проверяет задание. Критическая секция в коде представлена в методе `sitAtTheTableWithTeacher` класса `Teacher`. Таким образом, когда поток клиента попадает в критическую область, сервер, постоянно проверяя изменение текущего обрабатываемого клиента, замечает прибывший поток и начинает генерировать на него ответ, пока запрашиваемый поток ждёт в критической секции. Когда сервер обработал запрос и передал его клиенту, клиент выходит из критической секции, а сервер входит в режим ожидания. Также при создании потоков в программе был использован барьер (`pthread_barrier_t`), который позволяет начать работу потокам одновременно (по легенде задания получается, будто студенты вместе с преподавателем одновременно входят в аудиторию для экзамена).

Запускать программу необходимо с одним единственным аргументом – количеством прибывших на экзамен студентов, например:

```
./task05 10
```

Число интерфейсных модулей: 3, число файлов с определением программных объектов: 4. Общий размер исходных текстов: 8 037 байт, размер исполняемого файла: 29 128 байт.