

Отчет ИДЗ 4

Вариант 33

Выполнил Чугунов Андрей БПИ213

Условие

Пляшущие человечки. На тайном собрании глав преступного мира города Лондона председатель собрания профессор Мориарти постановил: отныне вся переписка между преступниками должна вестись тайнописью. В качестве стандарта были выбраны «пляшущие человечки», шифр, в котором каждой букве латинского алфавита соответствует хитроумный значок. Реализовать приложение, шифрующее исходный текст (в качестве ключа используется кодовая таблица, устанавливающая однозначное соответствие между каждой буквой и каким-нибудь числом). Каждый процесс-шифровальщик шифрует свои кусочки текста, многократно получаемые от менеджера. Распределение фрагментов текста между процессами-шифровальщиками осуществляется процессом-менеджером, который передает каждому процессу участок текста, получает от него результат, передает следующий незашифрованный фрагмент. Он же собирает из отдельных фрагментов зашифрованный текст. Количество процессов задается опционально. Каждый процесс может выполнять свою работу за случайное время. Сервер — процесс, распределяющий текст между шифровальщиками. Клиенты — процессы-шифровальщики

Реализация на 4-5

Порт сервера задается в командной строке. Там же задается кол-во клиентов. Далее с клавиатуры вводится строка, которую нужно закодировать. Каждый i -ый клиент должен кодировать все i -ые символы строки. Сервер создает сокет и ждет пока кто-то отправит туда данные. Когда, сервер получает данные, он получает нужные параметра клиента, для того чтобы общаться с ним по UDP. Соответственно сервер отправляет нужные символы тому, кто подключился. Когда клиент закодировал свои символы он отправляет их обратно серверу. Когда все клиенты закончили свою работу, сервер собирает исходную закодированную строку и выводит ее в консоль. Программа завершается когда все клиенты закончили шифрование

Реализация на 6-7

Логика программы остается той же. Добавляется программа state, которая получает состояние всей программы. Можно сказать, что state программа это очередной клиент, она подключается также как и клиенты. Отмечу, что первой должна подключаться именно state программа. Сервер общается с ней по протоколу UDP и отправляет свое состояние. Также клиент отправляет состояние серверу, чтобы в дальнейшем сервер отправил эти данные state программе. В итоге она получает состояние всей программы и выводит его в консоль. State программа работает, пока все клиенты не выполнят свои части. Когда клиенты закончили, сервер завершает работу, state тоже завершается.

Тесты

Скриншоты тестов программы лежат в репозитории в папке tests