

ИДЗ-4 Вариант 32

Лобанов Кирилл Сергеевич БПИ-213

Декабрь 2022

Условие задачи - Реализовать многопоточное приложение, шифрующее исходный текст (в качестве ключа используется кодовая таблица, устанавливающая однозначное соответствие между каждой буквой и каким-нибудь числом). Каждый поток шифрует свои кусочки текста. При решении использовалась парадигма портфеля задач. Файл `idz4.c` - программа на Си, `idz4` - исполняемая программа.

При решении использовался принцип "портфеля задач". Каждый поток при создании получил уникальный номер, с помощью которого он считал начало и конец подстроки данного текста, которую ему надо зашифровать.

4 балла) Была написана программа на языке C `idz4.c`, которая получает через консольный ввод текст любой длины, которая не превышает 100000, затем заменяет все буквы в нем на их двузначные коды. Коды генерируются в начале программы, они все различные.

5 баллов) В программу были добавлены комментарии, поясняющие роль создаваемых переменных и смысл выполняемых операций

Потоки в нашей программе создаются одним циклом и получают свой уникальный номер, затем в функции перевода текста поток вычисляет длину шифруемой подстроки и её начало в исходном тексте, а затем идет по ней, заменяет буквы на числа и записывает результат в итоговый массив. Благодаря уникальным номерам, все потоки имеют различные шифруемые подстроки, которые не пересекаются и покрывают весь исходный текст; отрезки, в которые потоки записывают ответ в итоговом массиве также не пересекаются и покрывают весь массив.

6-8 баллов) В начале программы создаются глобальные массивы для записи исходного текста, хранения ответа и ключей шифрования, глобальная переменная `buff_size` показывает длину текста. Затем создаются сами потоки, им присваиваются номера и они выполняют свою задачу по шифрованию текста. Затем выводится ответ, учитывая заглушки в итоговом массиве. Принцип заглушек описан в комментариях программы.

Добавлен ввод через командную строку, файловый ввод и рандомная генерация текста. при рандомном вводе надо вводить `./code --random file`, где `file` - это имя выходного файла

при консольном вводе надо вводить `./code --terminal file`, где `file` - это имя выходного файла

при файловом вводе надо вводить `./code --file file1 file2`, где `file1` - это имя входного файла, а `file2` - это имя выходного файла

при вводе из командной строки надо вводить `./code --terminal file word1 word2...`, где `file` - это имя выходного файла, `word1 word2...` - шифруемые слова

Для тестирования программы были использованы тесты, которые хранятся в папке `tests`.