

Тимохин Никита Игоревич

БПИ195

Вариант 27

27. 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 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 использовать парадигму портфеля задач.

Описание

При решении задачи использовалась парадигма портфеля задач. (каждый поток берет задачу из “Портфеля” и выполняет ее)

Источники:

http://staff.mmcs.sfedu.ru/~dubrov/files/tut_parallel_03_method.pdf

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

1. Запуск

При запуске программа предлагает ввести текст:



Работа программы при разных входных данных описана в файле «тестирование программы».

2. Описание работы

Для выполнения задачи с помощью нескольких потоков я делю введенный текст на подстроки. Каждый поток занимается своей частью текста (по 5 символов). Т.к. введенный текст может быть большим, а я создаю поток на каждые новые 5 символов я решил ограничить число потоков 1000. Т. к. использование большого кол-ва потоков нецелесообразно и только замедлит работу программы.

```
// Каждый поток обрабатывает по 5 символов (или меньше, если текст кончился), сразу создаю нужное кол-во потоков.  
int numOfThreads = ceil(inputStr.size() / 5.0);  
// При огромных текстах программа будет нецелесообразно создавать много потоков. Ограничим тысячей.  
if (numOfThreads > 1000) {  
    :  
    numOfThreads = 1000;  
}
```

Парадигма портфеля задач подразумевает использование какой-то переменной как портфель, из которого каждый поток выделяет задачу.

```
/// Портфель задач  
static int begin = 0;
```

В эту переменную записывается индекс начала работы следующего потока с введенной строкой. После того как поток забрал себе задачу(threadBegin) увеличиваем этот индекс.

```
threadBegin = begin;  
begin += 5;
```

И кодируем подстроку в соответствии с таблицей.

3. Генерация таблицы

В задании написано использовать таблицу буква-число. Я использую `map<char, char>`. Т.к. `char` это целочисленный тип, то заданию не противоречит. Таблица генерируется всегда одна и та же. В ней каждой букве латинского алфавита соответствует интересный символ.

Программа использует консольный ввод-вывод.

Разработка велась в MS VS 2019.

Тестирование и исходный код программы можно найти в файлах «Тестирование программы» и «openMP» соответственно.