Национальный исследовательский унивеситет
«Высшая школа экономики»

Факультет компьютерных наук
Департамент программной инженерии

Титульный лист

Задание на многопоточность. Портфель задач

Исполнитель
Студент БПИ198
Смирнов Михаил Алексеевич

Москва, 2020

1. Постановка задачи

Вариант 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.

2. Описание входных и выходных данных

На вход к программе подается название файла откуда считывать, название файла куда записывать результат, количество потоков (включая основной) и размер подзадач. Во входном файле должна быть строка состоящая только из латинский букв.

3. Описание алгоритма

Программа написана с использованием парадигмы портфеля задач [1]. Идея этого подхода в том, чтобы выделить такие задачи, которые может решать только один поток. То есть задача это неделимая единица. Изначально в “портфеле” лежать некоторые задачи. В ходе решения задач могут также возникать новые задачи. В программе должно быть заранее известно количество потоков (threads_count). В каждом потоке бесконечный цикл - если в “портфеле” есть задачи, то берем одну задачу и выполняем, иначе цикл завершен. Сами же задачи могут выполнятся разное время, однако обычно задач в несколько раз больше, чем потоков и суммарное время выполнения для потоков уравнивается. В общем каждый поток, включая основной, в моей программе делает примерно следующее:

```
void* do_shifr(void *params){
    while(true){
        task t = get_task();
        // Условие остановки
        if(t.size == 0) break;
        // Выполняет задачу
    }
    return nullptr;
}
```

Также нужно обратить внимание, что получение задач это критическая секция.

4. Тестирование

Неправильный путь к файлу.
Ввод: test.txt answer.txt 1 1
Вывод: File error

Неправильное число потоков.
Ввод: test1.txt answer1.txt -1 1
Вывод: Invalid threads_count

Все буквы.
Ввод: test1.txt answer1.txt 1 1

В файле: Qш |`»wM_Z▯rW+¤ЛU¬фЎ·▯hЁїдrDт§ёfi /?p]цх>уН}
R▯°И•Я

Во входном файле цифра.
Ввод: test2.txt answer2.txt 1 1
Вывод: Error: input string must contain only letters

Большой ввод, длина входной строки не делится нацело на размер заадач.
Ввод: test3.txt answer3.txt 1 99999
В файле: 10 млн. букв Q (хотя, возможно, зависит от кодировки).

Теперь тестирование многопоточности.

Один поток.
Ввод: test3.txt answer3.txt 1 100000
Вывод: 50ms

Несколько потоков.
Ввод: test3.txt answer3.txt 4 100000
Вывод: 24ms

Много потоков.
Ввод: test3.txt answer3.txt 100 100000
Вывод: 33ms

Маленький размер задач.
Ввод: test3.txt answer3.txt 4 1
Вывод: 1047ms

Большой размер задач.
Ввод: test3.txt answer3.txt 4 10000000
Вывод: 51ms

Т.е. количество потоков должно соответсвовать количеству ядер в процессоре, а размер задач не слишом маленьким, но и не слишком большим.

5. Источники

1. Портфель задач. https://l.wzm.me/_coder/custom/parallel.programming/003.htm (Пункт 3.6)