

Штанько Екатерина Олеговна

БПИ 193

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 решения

При решении использовалась парадигма портфеля задач (метод параллельных вычислений, при котором все задачи помещаются в портфель, разделяемый рабочими процессами. Каждый процесс многократно берет задачу из портфеля, выполняет ее и, возможно, порождает новые задачи, которые помещает в портфель. Вычисления завершаются, когда портфель пуст и рабочие процессы незаняты).

Источники:

<http://www.soft.architecture.net.ru/70/index-algoritmy-parallelnye-po-dannym-chast-8.htm>

https://l.wzm.me/_coder/custom/parallel.programming/003.htm

<http://edu.mmcs.sfedu.ru/file.php/74/OMPA1g2012.pdf>

Работая с потоками, я использовала информацию из лекций/семинаров, примеры кода с сайта <http://www.softcraft.ru/edu/comparch/practice/thread/03-openmp/>, а так же <http://ccfit.nsu.ru/arom/data/openmp.pdf>.

Прежде чем приступить к описанию выполненной работы, хочу обратить Ваше внимание на одну проблему, с которой я столкнулась.

Я работаю на Mac в CLion. С самого начала я столкнулась с проблемой подключения `#include <omp.h>`, и на ее решение ушел далеко не один час. В итоге, проблема решилась установкой libomp (`brew install libomp`) и внесением изменений в CMakeLists.txt. Это обеспечило мне возможно работать с omp, но `#include` выглядит в моем случае иначе, а именно `#include "/usr/local/opt/libomp/include/omp.h"`.

1. Запуск

Приложение запускается из командной строки.

Аргументы для запуска приложения из командной строки:

1. `argv[1]` - путь до файла для чтения входных данных. Если файла по данному пути не обнаружено - сообщение об ошибке:

```
Файл с входными данными найден не был!  
Process finished with exit code 0
```

2. `argv[2]` - путь до файла для записи выходных данных. Если файла по данному пути не обнаружено - создание нового файла.

3. argv[3] - число потоков. Требование:

1. Значение > 0! Если значение не соответствует - сообщение об ошибке:

```
Число потоков не может быть меньше 1!  
Process finished with exit code 0
```

2. Значение <=100! Если значение не соответствует - сообщение об ошибке:

```
Число потоков следует установить меньше <= 100!  
Process finished with exit code 0
```

(Хочу пояснить свое решение ограничить максимально возможное число потоков при вводе.

При тестировании программы и вводе очень больших значений в качестве аргумента кол-ва потоков возникала ошибка (а именно, ошибка начинала возникать при попытке ввести более чем 2048 потоков):

```
OMP: Error #34: System unable to allocate necessary resources for OMP thread:  
OMP: System error #35: Resource temporarily unavailable  
OMP: Hint Try decreasing the value of OMP_NUM_THREADS.
```

Попытка подстраховаться от возникновения подобной ошибки - первая причина ограничения.

Осознание того, что ввод большого числа потоков бессмыслен и лишь существенно замедляет работу программы - вторая причина.)

Если число аргументов не соответствует указанному - сообщение об ошибке:

```
Ошибка в данных переданных через командную строку!  
Process finished with exit code 0
```

2. Кодовая таблица

Соответствие устанавливается псевдослучайным образом с помощью srand(0). Символы выбираются из интервала от [129 до 166](исключая 152(начало строки) и 150(дублирует другой символ)) (ASCII) (они мне показались наиболее колоритными). Установленное соответствие хранится в словаре:

```
static std::map<char, int> letterSymbol;
```

4. Считывание из файла

(Код считывания из файла тривиален. Мне показалось неуместным его приводить здесь)
Программа способна шифровать как одну строку, так и последовательность строк, к примеру:

```
He doesn't care if it's in below  
He's sitting by the fire's glossy glow  
He don't care about the cold and the winds that blow  
He just says, let it snow, let it snow, let it snow  
Let it snow!  
1945
```

После считывания текста из файла я привожу все прописные латинские буквы к строчному виду.

5. Строка для записи результата

Для записи результата я создаю строку `outputString`, которую заполняю согласно следующему принципу:

- Если введена буква латинского алфавита - устанавливаю пробел
- Иначе - оставляю тот символ, что находился по этому индексу во входной строке

Таким образом, мне удастся сохранить, например, числа, символы или переходы на новую строку.

6. Работа с потоками и основная функция

Я постаралась качественно закомментировать код, описав в нем все теоретические аспекты, поэтому считаю уместным привести его:

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

void *func() {
    /// Под "словом" я подразумеваю последовательность с которой работает один
    поток

    /// Индекс начала одного "слова" в считанной последовательности
    int beginTask = 0;
    /// Вычисляем длину "слова" для одного потока
    int oneTaskStrLength = (int) ceil((double) (inputString.size()) /
numberOfThreads);
    /// Вычисляем кол-во "слов" в считанной последовательности
    int numberOfTasksInStr = (int) ceil((double) (inputString.size()) /
oneTaskStrLength);

    /// Область паралельного исполнения.
    /// Кол-во потоков задано через num_threads(numberOfThreads - 1).
    /// Я создаю на один поток меньше, чем было указано пользователем в командной
    строке.
    /// Делая так, я учитываю наличие основного потока.
    /// Условие private указывает на то, что каждый поток должен иметь свою копию
    переменной(beginTask - его задача) на всем протяжении своего исполнения.
    /// Условие shared указывает на то, что переменная(begin - портфель) будут
    разделяться между потоками. Все потоки будут доступаться к одной и той же
    области памяти.
    #pragma omp parallel num_threads(numberOfThreads - 1) private(beginTask)
    shared(begin)
    {
        /// Директива #pragma omp parallel for указывает на то, что данный цикл
        следует разделить по итерациям между потоками
        #pragma omp parallel for
        for (int i = 0; i < numberOfTasksInStr; ++i) {

            /// Критическая секция, в которой происходит выделение задачи из портфеля
            задач
            #pragma omp critical
            {
                beginTask = begin; /// (Выделение задачи)
                /// Перемещаем индекс начала на следующее слово
                begin += oneTaskStrLength;
            }

            if (beginTask < inputString.size()) {
                /// Выбираем из введенной последовательности "слово" для
                одного потока и кодируем его
                encoding(inputString.substr(beginTask, oneTaskStrLength),
beginTask);
            }
        }
    }
}
```

Используемый здесь метод **encoding** кодирует латинские символы в переданной строке и записывает результат в строку для вывода. Причем, метод устанавливает закодированный символ в результирующую строку именно на то место, на котором он находился в переданной строке.

Тестирование программы

Выше я уже описала и привела скрины работы программы в случае введенных некорректных данных.

Пример корректной работы программы:

В командную строку CLion введено:

/Users/miss.ekaterina/Desktop/test.txt /Users/miss.ekaterina/Desktop/answer.txt 3

До:

```
Oh the weather outside is frightful
But the fire is so delightful
Since we've no place to go
Let it snow, let it snow, let it snow

It doesn't show signs of stoppin'
And I've brought some corn for poppin'
The lights are turned down low
Let it snow, let it snow, let it snow

When we finally kiss goodnight
How I'll hate goin' out in the storm
But if you'll really hold me tight
All the way home I'll be warm

The fire is slowly dyin'
And, my dear, we're still goodbyin'
As long as you love me so
Let it snow, let it snow, let it snow

He doesn't care if it's in below
He's sitting by the fire's glossy glow
He don't care about the cold and the winds that blow
He just says, let it snow, let it snow, let it snow
Let it snow!
1945
```

После:

```
Ÿĸ Ĵĸ" "ĴĴĸ"ĥ Ÿ"ĴŸ...•" ...Ÿ ĥ...ĴĴĴ, "Ŵ
ĥ"Ĵ Ĵĸ" ĥ...ĥ" ...Ÿ ŸŸ •"Ŵ...ĴĴĴ, "Ŵ
Ÿ...," " " " " " " ,Ÿ ĸĴ," " ĴŸ ĴŸ
Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™, Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™, Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™

...Ĵ •Ÿ"Ÿ, 'Ĵ ŸĴŸ™ Ÿ...Ĵ, Ÿ Ÿ ŸŸŸ( (..., '
Ĵ, • ...' " " ĥĥŸ"ĴĴ ŸŸ—" 'Ÿĥ, ĥĥ (Ÿ( (..., '
Ĵĸ" Ŵ...ĴĴŸ Ĵĥ" Ĵ"ĥ, "• •Ÿ™, ŴŸ™
Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™, Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™, Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™

"ĴĴ", " " ĥ...ĴĴĴ Ĵ...ŸŸ ĴŸŸ•, ...ĴĴĴ
ĴŸ™ ...' ŴŴ ĴĴĴ" ĴŸ..., ' Ÿ"Ĵ ..., ĴĴ" ŸŸŸĥ—
ĥ"Ĵ ...' ĴŸ" 'ŴŴ ĥ"ĴĴĴĴ ĴŸŴ• —" Ĵ...ĴĴĴ
ĴŴŴ ĴĴ" "ĴĴ ĴŸ—" ...' ŴŴ ĥ" "ĴĴ—

ĴĴ" ĥ...ĥ" ...Ÿ ŸŴŸ™ŴĴ •Ĵ..., '
Ĵ, •, —Ĵ •"Ĵĥ, " " " " ŸĴ...ŴŴ ĴŸŸ•ĴĴ..., '
ĴŸ ŴŸ, Ĵ ĴŸ ĴŸ" ŴŸ" —" ŸŸ
Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™, Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™, Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™

Ĵ" •Ÿ"Ÿ, 'Ĵ 'Ĵĥ" ...' ...Ĵ' Ÿ ..., ĥ"ŴŸ™
Ĵ" 'Ÿ Ÿ...ĴĴ..., Ĵ ĴĴ ĴĴ" ĥ...ĥ" 'Ÿ ĴŴŸŸŸĴ ĴŴŸ™
Ĵ" •Ÿ, 'Ĵ 'Ĵĥ" ĴĥŸ"Ĵ ĴĴ" 'ŸŴ• Ĵ, • ĴĴ" " ..., •Ÿ ĴĴĴĴ ĴŴŸ™
Ĵ" "Ÿ"Ÿ ŸĴĴŸ, Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™, Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™, Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™
Ŵ"Ĵ ...Ĵ Ÿ,Ÿ™!
1945
```

Программа была протестирована и работает корректно как с 1 потоком, так и с большим числом потоков, например:

/Users/miss.ekaterina/Desktop/test.txt /Users/miss.ekaterina/Desktop/answer.txt 100

Еще один пример корректной работы программы, демонстрирующий:

1. Однозначность соответствия
2. Регистронезависимость

До:

aA bB cC dD eE fF gG hH iI jJ kK lL mM nN oO pP qQ rR sS tT uU vV wW xX yY zZ

После:

ќќ ґґ ‘ ‘ •• “” ’ ’ ъ ъ ќќ %‰ ГГ ҘҘ — , , ўў ‘ ‘ || ҺҺ ўў JJ “” ҘҘ ҘҘ ҺҺ ГГ ҺҺ

Еще один пример корректной работы программы, демонстрирующий возможность внедрения в текст чисел и знаков

До:

- Boss, we robbed a Bank! We took as much as\$ 100 - this is about 1% of what was in the Bank! :)
- It was attempt #1. You should try again...
- Call the number +79170467346?
- Uh... Well, why not?

После:

- ґўўў, “” Һўґґ”• ќ ґќ,Г! “” JўўГ ќў — ‘ќ ќў\$ 100 - Jќ...ў ...ў ґґў”J 1% ў, “ќќJ “ќў ..., Jќ” ґќ,Г! :)
- ...J “ќў ґJJ”-(J #1. Гў” ўґў”Ҙ• JґГ ґљќ...,...
- ‘ґҘҘ Jќ” , “-ґ”ґ +79170467346?
- “ќ... “”ҘҘ, “ќГ ,ўJ?

Также упомяну, что программа работает корректно, когда файл для считывания пуст.