

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Образовательная программа «Программная инженерия»

профессор департамента программной инженерии  
факультета компьютерных наук  
Легалов Александр Иванович

**Многопоточное приложение,  
дешифрующее кодированный текст**

Вариант: 28

**Отчет программы**

Исполнитель: студент группы \_ БПИ197  
Ф.И: Яхя Янал

**Москва 2020**

# Задание

И снова пляшущие человечки. Узнав о планах преступников озвученных в задаче 33, Шерлок Холмс предложил лондонской полиции специальную машину для дешифровки сообщений злоумышленников. Реализовать многопоточное приложение, дешифрующее кодированный текст. В качестве ключа используется известная кодовая таблица, устанавливающая однозначное соответствие между каждой буквой и каким-нибудь числом. Процессом узнавания кода в решении задачи пренебречь. Каждый поток дешифрует свои кусочки текста. При решении использовать парадигму портфеля задач.

# Решение

Для реализации задачи было использован взаимодействующие равные \_модель со способом распределение «портфель задач».

Сначала происходит чтение шифровку из файла, в функции `read_encrypted_message`, и потом вызывает `decoding` функцию с параметрами шифровка и количество необходимых потоков для выполнение все таски в задаче, где это число в программе равно корню длины шифровки(я знаю что количество подзадачей должно быть больше в 2-3 раза по минимум чем количество потоков, чтобы реально было эффективнее использовать парадигму «портфель задач», и знаю что каждая подзадача должна умеет какое то не низкое сложность выполнение , но чтобы было легче проверит программа я составил такое число, хотя вы можете менять его на другое логическое значение), и там в этом методе вызывается метод `make_tasks` чтобы разделить подзадачи и сохранить их в очередь, и после этого начинают потоки работают вместе на расшифрование шифровку ,где каждый поток берёт одну подзадачу, состоящую из части шифровки, и результат выполнение каждого потока сохраняется в определённой ячейкой в динамическим массиве `results` и потом собираются эти данные и выводится для пользователь на экране.

и так же в моей программе было реализовано класс `Timer` и `header files(Timer.h, Timer.hpp)` чтобы мы смогли читать время выполнение программы на любом таймфрейме.

## Вывод

Портфель задачи можно представить как в олимпиаде программирование где потоки эти людей в команде, и они получают какую то задачу, и им надо выполнить её на каком то время, если команда работала по принципу одному потоку, то есть один человек будет делать задачу, то скорее всего команда проиграет(если задача сложная или большая и делимая, иначе одного человек лучше для выполнения задачу), а если каждый участник команды взял одну подзадачу, и начал делать её, и потом соединить всё это вместе, то уже лучшее, но ещё если задача было разделено хорошо, чтобы все закончили почти вместе, а не один закончил свою подзадачу до второго в 30 мин раньше, то это ещё лучше и лучше, то есть просто этот способ распределение, может быть эффективный но не всегда.

# Почему выбрать OpenMP?

## Портативный

- Стандартизован для архитектур с общей памятью

## Просто и быстро

- Относительно легко распараллеливать небольшие части приложения за раз.
- Пошаговое распараллеливание
- Поддерживает как мелкозернистый, так и крупнозернистый параллелизм

## Компактный API

- Простой и ограниченный набор директив
- Не автоматическое распараллеливание

## Преимущества и недостатки

Преимущества	недостатки
<ul style="list-style-type: none"><li>• Общее адресное пространство обеспечивает удобство программирования</li><li>• Легкость программирования</li><li>• Обмен данными между потоками быстрый и единообразный (низкая задержка)</li><li>• Постепенное распараллеливание последовательного кода</li><li>• Оставляет управление потоками компилятору</li><li>• Непосредственно поддерживается компилятором</li></ul>	<ul style="list-style-type: none"><li>• Внутренние детали скрыты</li><li>• Программист несет ответственность за определение синхронизации, например замки</li><li>• Не может работать в распределенной памяти</li><li>• Производительность ограничена архитектурой памяти</li><li>• Отсутствие масштабируемости между памятью и процессорами.</li><li>• Требуется компилятор, поддерживающий OpenMP.</li><li>• Машины большего размера требуют большого веса</li></ul>

## Список используемых источников

- [https://l.wzm.me/\\_coder/custom/parallel.programming/003.htm](https://l.wzm.me/_coder/custom/parallel.programming/003.htm) раздел 3.6
- [http://window.edu.ru/catalog/pdf2txt/971/67971/41350?p\\_page=20](http://window.edu.ru/catalog/pdf2txt/971/67971/41350?p_page=20) § 4. Портфель задач СТ. 195-199
- Лекция 5. От ВШЭ Парадигмы параллельного программирования  
Параллельное программирование:  
[http://staff.mmcs.sfedu.ru/~dubrov/files/sl\\_parallel\\_05\\_paradigm.pdf](http://staff.mmcs.sfedu.ru/~dubrov/files/sl_parallel_05_paradigm.pdf)
- Видео уроки на YouTube Многопоточное программирование C++ :  
[https://www.youtube.com/watch?v=NawpxG81RRk&list=PLQOaTSbfxUtAc\\_RpyDiWCHq0YTzLtVSD0](https://www.youtube.com/watch?v=NawpxG81RRk&list=PLQOaTSbfxUtAc_RpyDiWCHq0YTzLtVSD0)
- <https://bisqwit.iki.fi/story/howto/openmp/>

## Ввод-Вывод данных

Формат входных данных организован с использованием файлов, в командной строке задаётся один параметр это путь на входной файл, где лежит шифровка.

Формат выходных данных происходит в консоли.

[illegible]

```
E:\Second Year HSE\AUS\HWS\Task04\Task04>cl /EHsc Timer.cpp /link /out:Task04.exe
Microsoft (R) C/C++ Optimizing Compiler Version 19.24.28314 for x64
Copyright (C) Microsoft Corporation. All rights reserved.

Task04.cpp
Timer.cpp
Generating Code...
Microsoft (R) Incremental Linker Version 14.24.28314.0
Copyright (C) Microsoft Corporation. All rights reserved.

/out:Task04.exe
/out:Task04.exe
Task04.obj
Timer.obj

E:\Second Year HSE\AUS\HWS\Task04\Task04>ls
README.md Task04.cpp Task04.exe Task04.obj Timer.cpp Timer.h Timer.hpp Timer.obj input
E:\Second Year HSE\AUS\HWS\Task04\Task04>Task04 input/test1.txt
===== NUMBER OF WORKING PROCESSES =====
8

===== ENCRYPTED MESSAGE =====
441910173135296338151028151114351030251411351011301015191718301019241030181510153215241924176359151011281510172519241710
302510282512102912152812112421

===== DECODED MESSAGE =====
Hi guys
Be ready today at eight in the evening
We are going to rob sberbank

===== RUNTIME =====
44554 milliseconds

E:\Second Year HSE\AUS\HWS\Task04\Task04>
```

```
E:\Second Year HSE\AUS\HWs\Task04_Task04>Task04 input/test2.txt
===== NUMBER OF WORKING PROCESSES =====
```

```
4
```

```
===== ENCRYPTED MESSAGE =====
```

```
3925232811141529103319301814281133
```

```
===== DECODED MESSAGE =====
```

```
Comrades withdraw
```

```
===== RUNTIME =====
```

```
12083 milliseconds
```

```
===== NUMBER OF WORKING PROCESSES =====
```

```
24
```

```
===== ENCRYPTED MESSAGE =====
```

```
441910173135296338151028151114351030251411351011301015191718301019241030181510153215241924176359151011281510172519241710
302510282512102912152812112421634419101731352963381510281511143510302514113510113010151917183010192410301815101532152419
241763591510112815101725192417103025102825121029121528121124216344191017313529633815102815111435103025141135101130101519
171830101924103018151015321524192417635915101128151017251924171030251028251210291215281211242163441910173135296338151028
151114351030251411351011301015191718301019241030181510153215241924176359151011281510172519241710302510282512102912152812
112421634419101731352963381510281511143510302514113510113010151917183010192410301815101532152419241763591510112815101725
192417103025102825121029121528121124216344191017313529633815102815111435103025141135101130101519171830101924103018151015
321524192417635915101128151017251924171030251028251210291215281211242163441910173135296338151028151114351030251411351011
301015191718301019241030181510153215241924176359151011281510172519241710302510282512102912152812112421634419101731352963
381510281511143510302514113510113010151917183010192410301815101532152419241763591510112815101725192417103025102825121029
1215281211242163
```

```
===== DECODED MESSAGE =====
```

```
Hi guys
Be ready today at eight in the evening
We are going to rob sherbank
Hi guys
Be ready today at eight in the evening
We are going to rob sherbank
Hi guys
Be ready today at eight in the evening
We are going to rob sherbank
Hi guys
Be ready today at eight in the evening
We are going to rob sherbank
Hi guys
Be ready today at eight in the evening
We are going to rob sherbank
Hi guys
Be ready today at eight in the evening
We are going to rob sherbank
Hi guys
Be ready today at eight in the evening
We are going to rob sherbank
Hi guys
Be ready today at eight in the evening
We are going to rob sherbank
Hi guys
Be ready today at eight in the evening
We are going to rob sherbank
```

```
===== RUNTIME =====
```

```
129510 milliseconds
```



[illegible]