Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ» (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

ОТЧЕТ

по лабораторной работе № 5

Название: Конвейер

Дисциплина: Анализ алгоритмов

Студент ИУ7-52Б _____ Короткая В. М.
(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ Волкова Л.Л.
(Подпись, дата) (И.О. Фамилия)

Москва, 2021

Содержание

Введение

КОНВЕЙЕР - (англ. conveyer, от convey – продвигать, перевозить) (транспортёр), машина непрерывного действия, служащая для перемещения сыпучих, кусковых, штучных и др. грузов.

Историческая справка

Машины непрерывного действия были известны уже в глубокой древности – за неск. тысячелетий до н. э. Подъёмники с ковшами применяли для водоснабжения и в системах орошения, при строительстве укреплений, на рудниках. В Древнем Египте и в Китае использовали водоподъёмные машины, а также устройства с толкающими скребками, винтом. В Зап. Европе в 16–17 вв. дерев. винтовые К. устанавливали на мукомольных предприятиях.

Разработчики архитектуры компьютеров издавна прибегали к методам проектирования, известным под общим названием "совмещение операций при котором аппаратура компьютера в любой момент времени выполняет одновременно более одной базовой операции. Этот общий метод включает два понятия: параллелизм и конвейеризацию. Параллелизмы был рассмотрен в предыдущей работе. Сейчас время конвейера.

Цель данной работы - разработка и исследование конвейерных вычислений.

Для достижения данной цели необходимо рассмотреть следующие задачи:

- описать конвейерную обработку и применение ее на практики;
- реализовать конвейерную обработку на примере шифрования строк;
- провести экспериментальные замеры времени реализованного конвейера.

1. Аналитическая часть

В данном разделе рассматривается понятие конвейерной обработки. Так же рассматривается описание решаемой задачи в лабороторной работе.

1.1. Конвейерная обработка

Конвейеризация (или конвейерная обработка) в общем случае основана на разделении подлежащей исполнению функции на более мелкие части, называемые ступенями, и выделении для каждой из них отдельного блока аппаратуры. Так обработку любой машинной команды можно разделить на несколько этапов (несколько ступеней), организовав передачу данных от одного этапа к следующему. При этом конвейерную обработку можно использовать для совмещения этапов выполнения разных команд. Производительность при этом возрастает благодаря тому, что одновременно на различных ступенях конвейера выполняются несколько команд. Конвейерная обработка такого рода широко применяется во всех современных быстродействующих процессорах.

1.2. Алгоритмы шифрования строк

Идея шифрования строк - преобразование их для скрытия от посторонних. В то же время, те которым предназначается информация способны ее дешифровать.

В лабораторной работе будут реализованы шифры Цезаря и Вернама (XOR - шифр).

Шифр Цезаря - имеется ключ в виде числа от 1 до 25 (для латиницы) и каждая буква строки смещается на значение равное ключу.

Шифр Вернама (XOR - шифр) - строка разбивается на отдельные символы и каждый символ представляется в бинарном виде. Далее с каждым символом применяется операция XOR с ключом. В итоге получаем зашифрованную строку.

Вывод

В данном работе стоит задача реализации ассинхронных конвейерных вычислений. Были рассмотрены особенности построения конвейерных вычис-

лений.

Входными данными является строка.

Выходными данными является зашифрованная строка.

На программу накладываюся следующие ограничения:

- строка не должна быть пустой.

2. Конструкторска часть

В данном разделе будут приведены схемы алгоритмов шифрования и модель вычислений.

2.1. Схемы

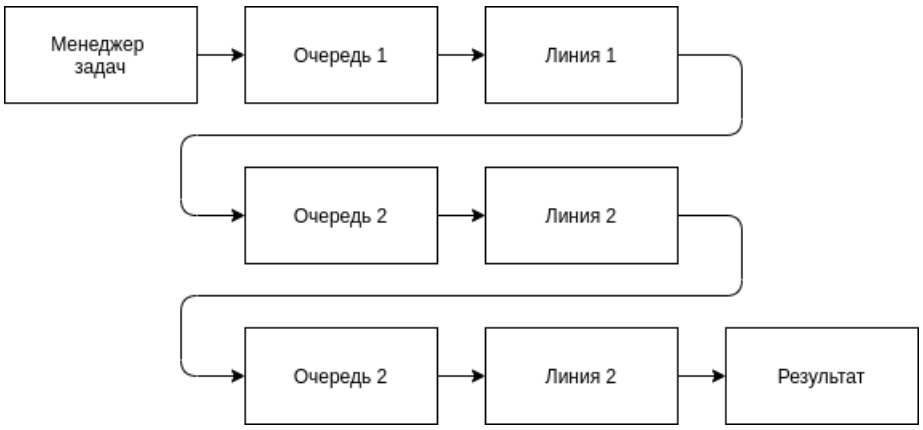

Рис. 2.1: Схема организации конвейерных вычислений.

2.2. Структура ПО

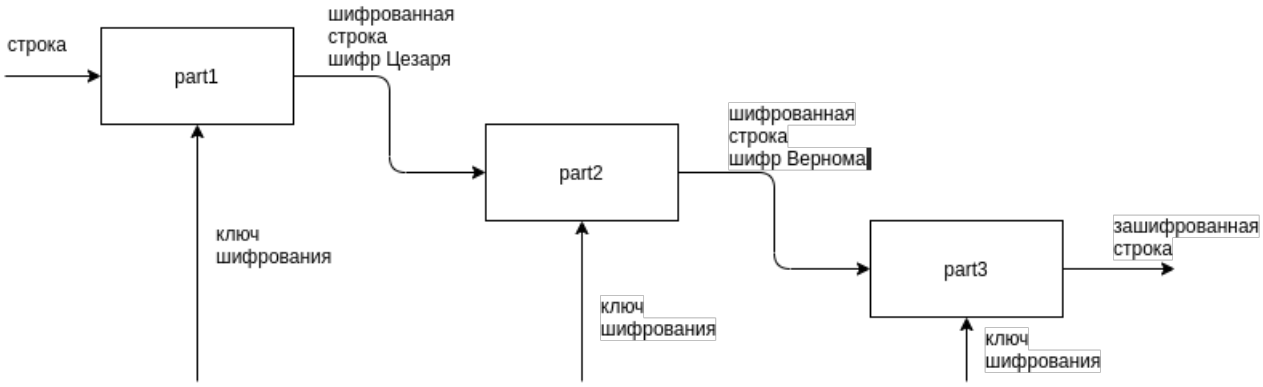

Рис. 2.2: idef0 - диограмма конвейера.

2.3. Схемы

Для реализации конвейерных вычислений, введем пользовательские типы данных:

- struct Request - описывает результат обработки задачи на конвейерной ленте

```
1 struct Request{
2     double timeS[3];
3     double timeE[3];
4
5     std::string dataStr;
6     int len, num;
7 };
```

Здесь:

- timeS[3] - массив, содержащий время начала обработки строки на каждой ленте;
- timeE[3] - массив, содержащий время окончания обработки строки на каждой ленте;
- dataStr - строка, переданная для обработки;
- len - длина строки;
- num - номер заявки.

Вывод

На основании теоритических данных была построена схема организации конвейерных вычислений на примере конвейера с тремя лентами.

Так же приведено описание пользовательских структур данных.

3. Технологическая часть

В данном разделе приведены средства реализации, требования к ПО и листинги кода.

3.1. Средства реализации

В качестве языка программирования был выбран c++. Данный язык знаком и предостовляет все необходимые ресурсы. В качестве среды разработки я использовала Visual Studio Code, т.к. считаю его достаточно удобным и легким. Visual Studio Code подходит не только для Windows, но и для Linux, это еще одна причина, по которой я выбрала VS code, т.к. у меня установлена ОС fedora 34.

3.2. Сведенья о модулях программы

Данноя программа разбита на модули:

- main.cpp - файл, содержащий точку входа в программу;
- conveyor.cpp - файл, содержащий реализацию конвейера.

3.3. Реализация конвейера

Листинг 3.1: Функция запускающая конвейер.

```
1  void Conveyor::run()
2  {
3      generateRequest();
4
5      std::thread t1 = std::thread(&Conveyor::part1, this);
6      std::thread t2 = std::thread(&Conveyor::part2, this);
7      std::thread t3 = std::thread(&Conveyor::part3, this);
8
9      t1.join();
10     t2.join();
11     t3.join();
12 }
```

Листинг 3.2: 1 Лента - шифр Цезаря.

```
1  void Conveyor::part1()
2  {
3      for(; ft1 < ntask; ft1++)
4      {
5          Request *req;
6
7          if (startQ.size())
8          {
9              req = startQ.front();
10             startQ.pop();
11         }
12         else continue;
13
14         req->timeS[0] = clock();
15         req->cryptCaesar(12);
16         req->timeE[0] = clock();
17
18         m1.lock();
19         q2.push(req);
20         m1.unlock();
21     }
22 }
```

Листинг 3.3: 2 Лента - шифр Вернама.

```
1  void Conveyor::part2()
2  {
3      while(q2.size() == 0)
4          continue;
5
6      for (; ft2 < ntask; ft2++)
7      {
8          while(q2.size() == 0)
9              continue;
10
```

```
11          Request *req;
12
13          m1.lock();
14
15          req = q2.front();
16          q2.pop();
17
18          m1.unlock();
19
20          req->timeS[1] = clock();
21          req->cryptXor('p');
22          req->timeE[1] = clock();
23
24          m2.lock();
25          q3.push(req);
26          m2.unlock();
27      }
28  }
```

Листинг 3.4: 3 Лента - шифр Вернама..

```
1  void Conveyor::part3()
2  {
3      while(q3.size() == 0)
4          continue;
5
6      for(; ft3 < ntask; ft3++)
7      {
8          while (q3.size() == 0)
9          continue;
10
11         Request *req;
12
13         m2.lock();
14
15         req = q3.front();
```

```
16              q3.pop();
17
18              m2.unlock();
19
20              req->timeS[2] = clock();
21              req->cryptXor('a');
22              req->timeE[2] = clock();
23
24              result.push_back(req);
25          }
26  }
```

Вывод

В данном разделе был реализован выше описанный алгоритм шифрования. Разработано программное обеспечение, представленны листинги программы.

4. Исследовательская часть

В данном разделе будет произведено измерение временных характеристик.

4.1. Технические характеристики

Технические характеристики устройства на котором выполнялось исследование:

- процессор Intel® Core™ i5-10210U CPU @ 1.60GHz × 8;
- память 15.3 GiB;
- операционная система Fedora 34 (Workstation Edition) 64-bit.

Исследования проводились на ноутбуке включенном в сеть электропитания. Во время тестирования ноутбук был нагружен приложениями окружения рабочего стола, а так же системой тестирования.

4.2. Временные характеристики

Замеры времени проводились на конвейере, обрабатывающем 200 заявок, каждая из которых содержит строку длинной 1000000 символов. Было замерено время 20 конвейеров, приведены усреднённые результаты.

На графиках 4.1 - 4.2 представлены результаты замеров времени во 2 и 3 очередях, и общее время, проведённое заявкой во всей системе, где t в миллисекундах.

Из графиков видно, что время нахождения во второй очереди резко растёт только в начале обработки, но затем спадает и держится на низком уровне. В третьей очереди и всей системе наблюдается такая же картина, однако пики выше.

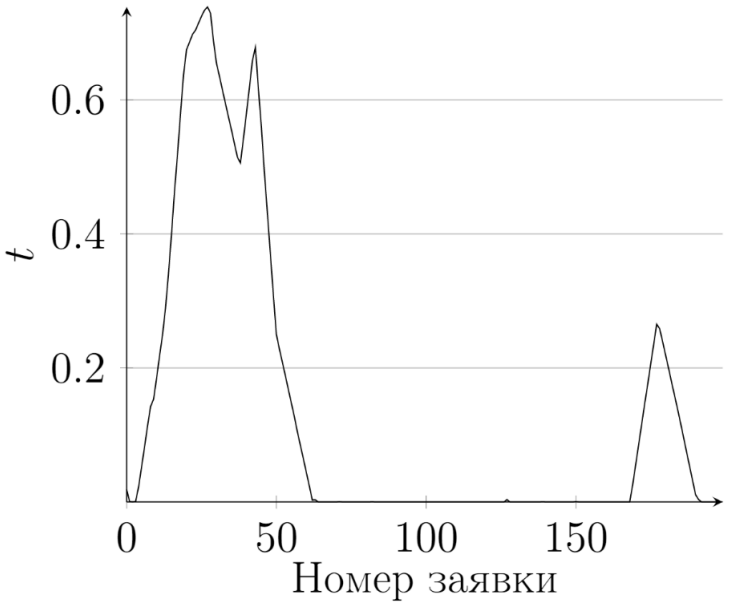

Рис. 4.1: время проведённое заявкой во второй очереди

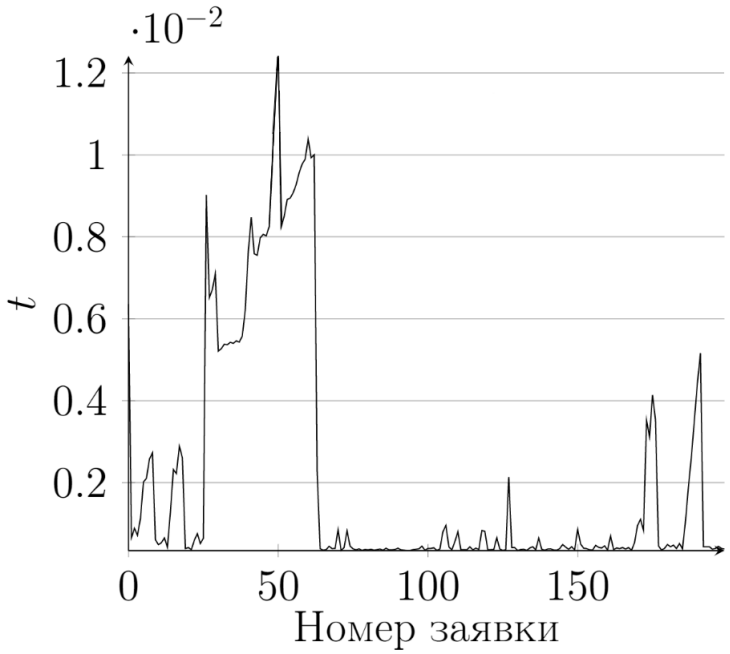

Рис. 4.2: время проведённое заявкой в третьей очереди

В следующей таблице 4.1 приведены минимальные, максимальные и средние результаты замеренного времени в миллисекундах, проведённого заявками в очередях и во всём конвейере.

Можно сказать, что наибольшее время потрачено в ожидании поступления на конвейер, а значит первый этап является наиболее затратным по времени.

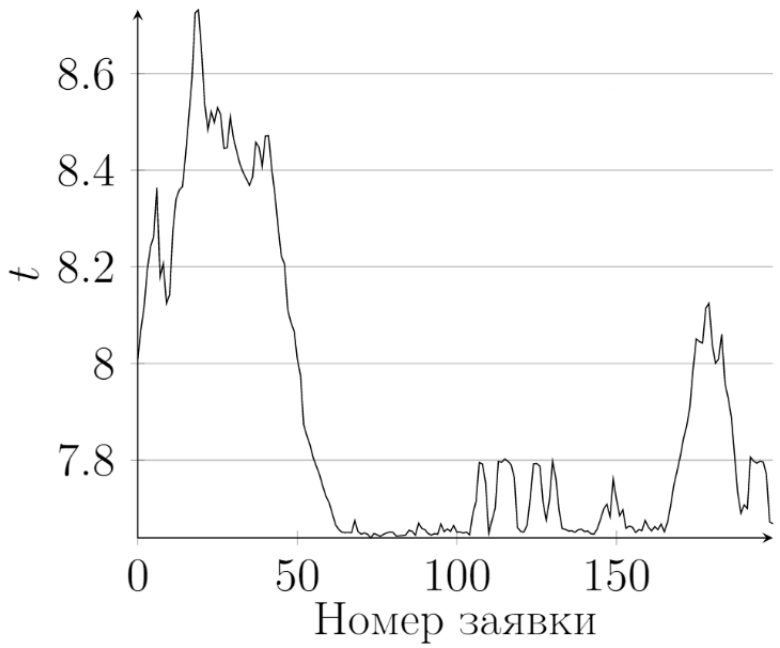

Рис. 4.3: время проведённое заявкой в конвейере

Таблица 4.1: результаты проведённого заявками времени в очередях и системе

	min	max	average
Вторая очередь	$3 \cdot 10^{-4}$	0.739	0.140
Третья очередь	$3 \cdot 10^{-3}$	0.013	0.002
Система	7.753	8.732	7.890

Вывод

В данном разделе были рассмотрены результаты работы программы. Из анализа стало ясно, что первый этап - шифрование с помощью шифра Цезаря замедляет работу всей системы, а также, что разница во времени работы 2 и 3 этапов крайне мала, что следует из малого времени, проведённого в третьей очереди.

Заключение

В ходе лабораторной работы достигнута поставленная цель: разработка и исследование конвейерных вычислений и использование их на практике. Также решены все поставленные задачи.

Стало ясно, что шифрование Цезаря занимает достаточно большую часть времени обработки заявки, в то время как два XOR шифра выполняются с равной скоростью. Также стало ясно, что разделение основной задачи на этапы даёт положительный результат на общем времени работы программы.

Список литературы

1. Дж. Макконнел. Анализ алгоритмов. Активный обучающий подход. – М.: Техносфера, 2017. – 267с.

2. Основы программирования на языках Си и C++ для начинающих[Электронный ресурс]. Режим доступа: http://cppstudio.com/ (дата обращения 10.10.2021)

3. LINUX.ORG.RU - Русскоязычная информация о ОС Linux[Электронный ресурс] Режим доступа://www.linux.org.ru/(дата обращения 25.10.2021)

4. Документация языка C++ 98 [Электронный ресурс], режим доступа: http://www.open-std.org/JTC1/SC22/WG21/ (дата обращения 10.12.2021)

5. Р.А. Волков, А.Н. Гнутов, В.К. Дьячков и др. Конвейеры. Справочник. -Машиностроение, Ленинградское отделение, 1984.