



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»

Отчёт по лабораторной работе №4 по дисциплине "Анализ алгоритмов"

Тема Параллельное умножение матриц

Студент Романов А.В.

Группа ИУ7-53Б

Оценка (баллы) _____

Преподаватели Волкова Л.Л., Строганов Ю.В.

Оглавление

Введение	2
1 Аналитическая часть	3
1.1 Стандартный алгоритм	3
1.2 Алгоритм Копперсмита – Винограда	3
1.3 Вывод	3
2 Конструкторская часть	4
2.1 Схемы алгоритмов	4
2.2 Вывод	4
3 Технологическая часть	6
3.1 Требование к ПО	6
3.2 Средства реализации	6
3.3 Реализация алгоритмов	6
3.4 Тестовые данные	8
3.5 Вывод	8
4 Исследовательская часть	9
4.1 Технические характеристики	9
4.2 Время выполнения алгоритмов	9
4.3 Вывод	9
Заключение	10
Литература	10

Введение

Задачи лабораторной работы:

1. test

1 | Аналитическая часть

1.1 Стандартный алгоритм

1.2 Алгоритм Копперсмита – Винограда

1.3 Вывод

Вывод

2 | Конструкторская часть

2.1 Схемы алгоритмов

Рис. 2.1: Схема стандартного алгоритма умножения матриц

Рис. 2.2: Схема алгоритма Копперсмита – Винограда

Рис. 2.3: Схема функций алгоритма Копперсмита – Винограда

2.2 Вывод

На основе теоретических данных, полученных из аналитического раздела, были построены схемы обоих алгоритмов умножения матриц. Оценены их трудоёмкости в лучшем и худшем случаях.

Рис. 2.4: Схема функций алгоритма Копперсмита – Винограда

Рис. 2.5: Схема оптимизированного алгоритма Копперсмита – Винограда

Рис. 2.6: Схема функций оптимизированного алгоритма Копперсмита – Винограда

3 | Технологическая часть

В данном разделе приведены средства реализации и листинги кода.

3.1 Требование к ПО

К программе предъявляется ряд требований:

- 123

3.2 Средства реализации

Для реализации ПО я выбрал язык программирования Haskell [?]. Данный выбор обусловлен ???

3.3 Реализация алгоритмов

В листингах 3.1 - 3.4 приведена реализация алгоритмов перемножения матриц.

Листинг 3.1: Функция умножения матриц обычным способом

```
1 void base_multiplication(args_t *args) {  
2     for (int i = 0; i < N; i++) {  
3         for (int j = 0; j < K; j++) {  
4             args->res[i][j] = 0;  
5             for (int k = 0; k < M; k++) {  
6                 args->res[i][j] += args->m1[i][k] * args->m2[k][j];  
7             }  
8         }  
9     }  
10 }
```

Листинг 3.2: Функция умножения матриц параллельно. Способ №1

```
1 void *parallel_multiplication1(void *args) {  
2     pthread_args_t *argsp = (args_t *)args;  
3  
4     int row_start = argsp->tid * (argsp->size / argsp->cnt_threads);  
5     int row_end = (argsp->tid + 1) * (argsp->size / argsp->cnt_threads);  
6 }
```

```

7   for (int i = row_start; i < row_end; i++) {
8       for (int j = 0; j < K; j++) {
9           argsp->mult_args->res[i][j] = 0;
10          for (int k = 0; k < M; k++) {
11              argsp->mult_args->res[i][j] += argsp->mult_args->m1[i][k] * argsp->
                  mult_args->m2[k][j];
12          }
13      }
14  }
15
16  return NULL;
17 }

```

Листинг 3.3: Функция умножения матриц параллельно. Способ №2

```

1  void *parallel_multiplication2(void *args) {
2      pthread_args_t *argsp = (args_t *)args;
3
4      int col_start = argsp->tid * (argsp->size / argsp->cnt_threads);
5      int col_end = (argsp->tid + 1) * (argsp->size / argsp->cnt_threads);
6
7      for (int i = 0; i < N; i++) {
8          for (int j = col_start; j < col_end; j++) {
9              argsp->mult_args->res[i][j] = 0;
10             for (int k = 0; k < M; k++) {
11                 argsp->mult_args->res[i][j] += argsp->mult_args->m1[i][k] * argsp->
                        mult_args->m2[k][j];
12             }
13         }
14     }
15
16     return NULL;
17 }

```

3.4 Тестовые данные

В таблице 3.1 приведены тесты для функций, реализующих параллельное и обычное умножение матриц. Все тесты пройдены успешно.

3.5 Вывод

Вывод 3

Первая матрица	Вторая матрица	Ожидаемый результат
$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	$\begin{pmatrix} 6 & 12 & 18 \\ 6 & 12 & 18 \\ 6 & 12 & 18 \end{pmatrix}$
$\begin{pmatrix} 1 & 2 & 2 \\ 1 & 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 2 \end{pmatrix}$	$\begin{pmatrix} 5 & 10 \\ 5 & 10 \end{pmatrix}$
(2)	(2)	(4)
$\begin{pmatrix} 1 & -2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	$\begin{pmatrix} -1 & 2 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}$	$\begin{pmatrix} 0 & 4 & 6 \\ 4 & 12 & 18 \\ 4 & 12 & 18 \end{pmatrix}$

Таблица 3.1: Тестирование функций

4 | Исследовательская часть

4.1 Технические характеристики

Ниже приведены технические характеристики устройства, на котором было проведено тестирование ПО:

- Операционная система: Debian [3] Linux [4] 11 «bullseye» 64-bit.
- Оперативная память: 12 GB.
- Процессор: Intel(R) Core(TM) i5-3550 CPU @ 3.30GHz [5].

4.2 Время выполнения алгоритмов

Тики

В таблицах 4.1 и 4.2 представлены замеры времени работы для

Таблица 4.1: Таблица времени выполнения алгоритмов (в секундах)

Размер матрицы	C	ТС	KB	OKB
100	0.482	0.120	0.169	0.158
200	3.998	0.947	1.312	1.253
300	13.526	3.364	4.478	4.063
400	34.330	7.521	11.319	9.909
500	NaN	14.913	22.427	19.357
600	NaN	26.448	39.292	33.265

4.3 Вывод

Заключение

В рамках данной лабораторной работы:

1. тест

Литература

- [1] Реализация алгоритма умножения матриц по Винограду на языке Haskell. Анисимов Н.С, Строганов Ю.В. [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/realizatsiya-algoritma-umnozheniya-matrits-po-vinogradu-na-yazyke-haskell/viewer>. Дата обращения: 01.10.2020.
- [2] The Haskell purely functional programming language [Электронный ресурс]. Режим доступа: <https://haskell.org/>. Дата обращения: 16.09.2020.
- [3] Debian – универсальная операционная система [Электронный ресурс]. Режим доступа: <https://www.debian.org/>. Дата обращения: 20.09.2020.
- [4] Linux – Getting Started [Электронный ресурс]. Режим доступа: <https://linux.org>. Дата обращения: 20.09.2020.
- [5] Процессор Intel® Core™ i5-3550 [Электронный ресурс]. Режим доступа: <https://ark.intel.com/content/www/ru/ru/ark/products/65516/intel-core-i5-3550-processor-6m-cache-up-to-3-70-ghz.html>. Дата обращения: 20.09.2020.
- [6] Haskell profiling [Электронный ресурс]. Режим доступа: https://downloads.haskell.org/ghc/8.8.1/docs/html/user_guide/index.html. Дата обращения: 20.09.2020.