Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное 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 **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

Отчет
по лабораторной работе № 4

по дисциплине «Организация ЭВМ и систем»

Тема: «Организация памяти конвейерных суперскалярных ЭВМ»

Студент группы ИУ6-73Б _____ В.К. Залыгин
(Подпись, дата) (И.О. Фамилия)

Преподаватель _____ М.П. Калитвенцев
(Подпись, дата) (И.О. Фамилия)

2025 г.

Цель работы

Освоение принципов эффективного использования подсистемы памяти современных универсальных ЭВМ, обеспечивающей хранение и своевременную выдачу команд и данных в центральное процессорное устройство. Работа проводится с использованием программы для сбора и анализа производительности PCLAB.

В ходе работы необходимо ознакомиться с теоретическим материалом, касающимся особенностей функционирования подсистемы памяти современных конвейерных суперскалярных ЭВМ, изучить возможности программы PCLAB, изучить средства идентификации микропроцессоров, провести исследования времени выполнения тестовых программ, сделать выводы о архитектурных особенностях используемых ЭВМ.

Выполнение работы

Эксперимент 1 «*Исследования расслоения динамической памяти*»

Исходные данные эксперимента: по полученной характеристике определены значения T1 = 640 и координата экстремума T2 = 32768 по оси X. Принята длина строки кэш-памяти П = 64 байта, объём оперативной памяти О = $4 \cdot 2^{30}$ байт. По результатам расчётов получено число банков Б = T1/П = 10, оценка размера страницы РС = T2/Б = 3277, а также оценка количества страниц С = О/(РС·Б·П) = 2048.

Настраиваемые параметры: шаг приращения адреса читаемых данных (ось X) и единицы измерения по оси Y (время в мкс либо число тактов). Снимок графика временной характеристики представлен на рисунке 1, где красная кривая отражает время (или число тактов) выполнения при последовательном чтении адресов с заданным шагом.

Результат эксперимента: по положению экстремума и расчётам оценены параметры организации памяти (банки/страницы) и выявлено, что время доступа заметно зависит от шага адресации (из-за особенностей трансляции адресов и распределения по банкам/страницам).

Вывод: структура адресации (шаг чтения) может создавать регулярные «невыгодные» режимы, приводящие к росту времени доступа, что позволяет по экспериментальной кривой оценивать параметры расслоения/страничной организации памяти.

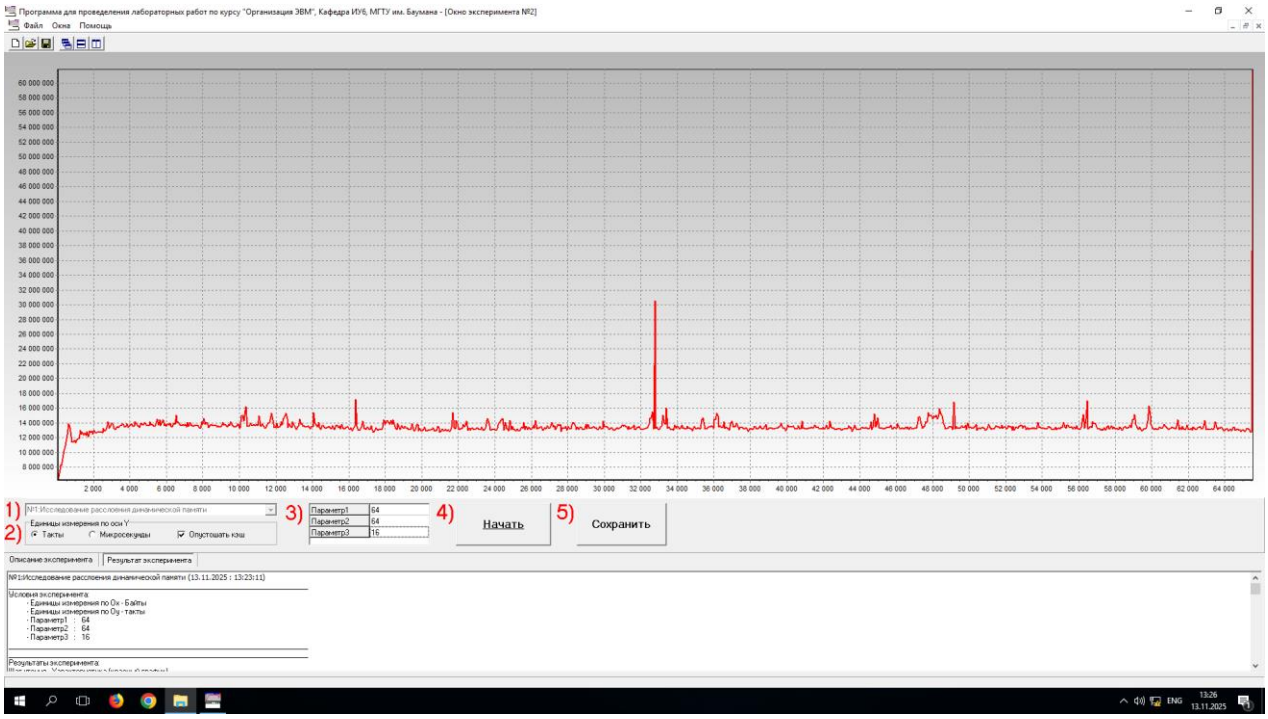

Рисунок 1 – Результат 1ого эксперимента

На рисунке 2 приведены описания параметров эксперимента.

Параметр	Диапазон	Масштаб	Описание
№ 1	1..128	К	Максимальное расстояния между читаемыми блоками
№ 2	4..64	Б	Шаг увеличения расстояния между читаемыми 4-х байтовыми ячейками.
№ 3	1..16	М	Размер массива

Рисунок 2 – Описание параметров 1ого эксперимента

Эксперимент 2 «*Сравнение эффективности ссылочных и векторных структур*»

Исходные данные эксперимента: сравнивались два варианта доступа к данным — алгоритм на списке (ссылочная структура) и алгоритм на массиве (векторная структура). Настраиваемый параметр эксперимента — фрагментация списка (ось X), а измеряемая характеристика — время выполнения (или число тактов) по оси Y. Графики результатов приведены на рисунке 3, где красная линия соответствует списку, а зелёная — массиву.

Результат эксперимента: время работы алгоритма со списком существенно выше и сильнее зависит от фрагментации, тогда как алгоритм с массивом демонстрирует более стабильное и низкое время выполнения.

Вывод: непрерывное размещение данных (массив) лучше использует кэш и предвыборку, а ссылочные структуры при росте фрагментации ухудшают локальность и приводят к росту задержек из-за более частых промахов кэша и обращений к памяти.

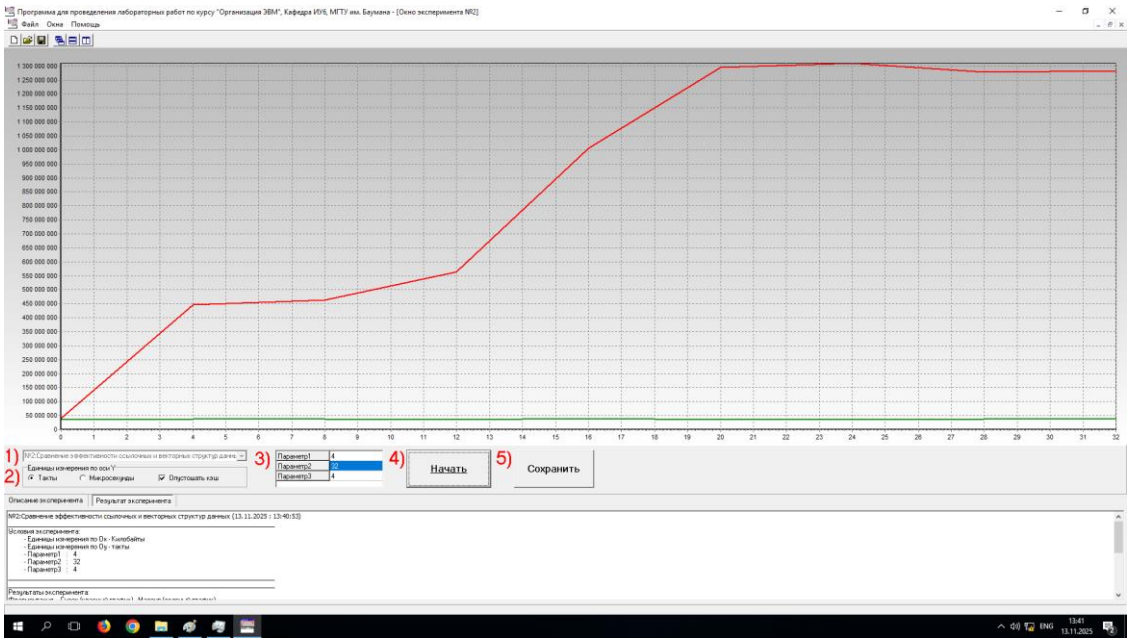

Рисунок 3 – Результат 2ого эксперимента

На рисунке 4 приведены описания параметров эксперимента.

Параметр	Диапазон	Масштаб	Описание
№ 1	1..20	M	Количество элементов в списке
№ 2	4..500	K	Максимальная фрагментации списка
№ 3	1..10	K	Шаг увеличения фрагментации

Рисунок 4 – Описание параметров 2ого эксперимента

Эксперимент 3 «*Исследование эффективности программной предвыборки*»

Исходные данные эксперимента: сравнивались два режима выполнения алгоритма чтения — без предвыборки и с программной предвыборкой. Настраиваемый параметр — смещение читаемых данных от начала блока (ось X), измеряемая характеристика — время выполнения или число тактов (ось Y, в зависимости от выбранных единиц). Графическое сравнение режимов представлено на рисунке 5: красная кривая соответствует режиму без предвыборки, зелёная — с предвыборкой.

Результат эксперимента: при использовании предвыборки достигается уменьшение времени выполнения относительно базового варианта, что особенно заметно на диапазонах смещений, где обращение к памяти иначе приводило бы к ожиданию данных.

Вывод: программная предвыборка позволяет частично скрывать латентность памяти за счёт раннего запроса данных, повышая эффективность чтения при регулярных паттернах доступа.

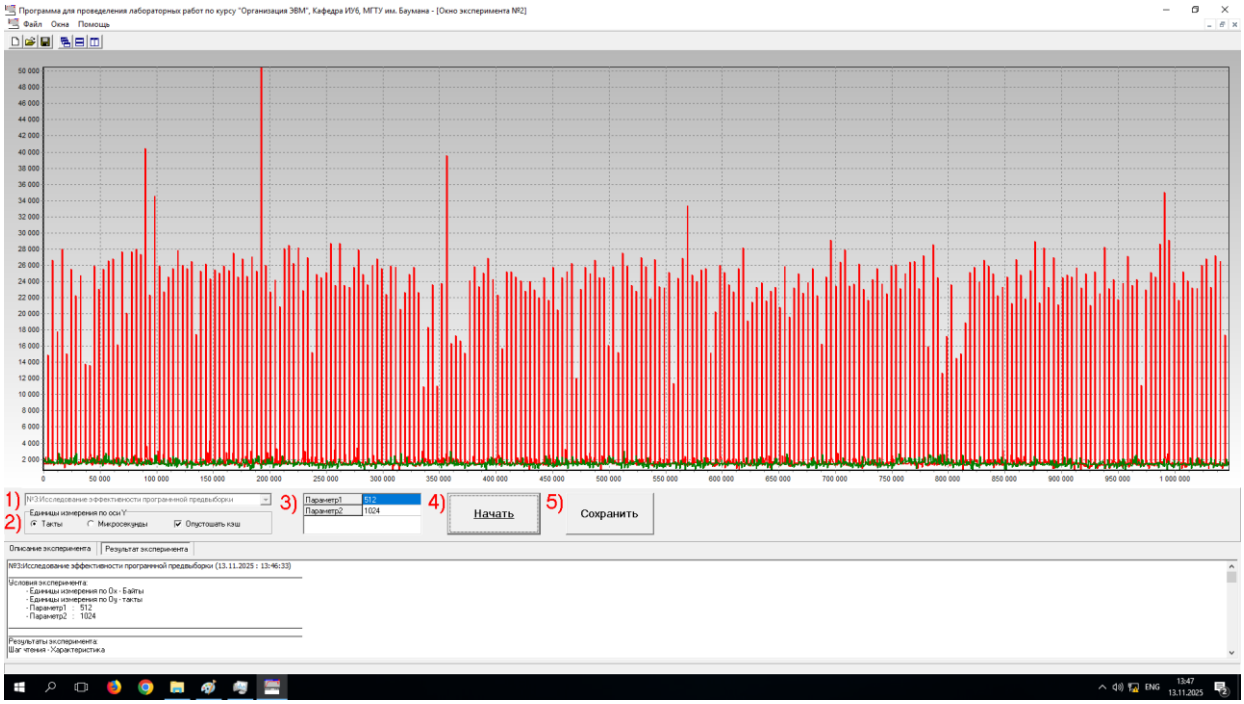

Рисунок 5 – Результат 3ого эксперимента.

На рисунке 6 приведены описания параметров эксперимента.

№ 1	1..4096	Б	Шаг увеличения расстояния между читаемыми данными
№ 2	4..8192	К	Размер массива

Рисунок 6 – Описание параметров 3ого эксперимента

Эксперимент 4 «*Исследование способов эффективного чтения оперативной памяти*»

Исходные данные эксперимента: сравнивались два варианта структур данных при обработке нескольких массивов — неоптимизированная структура (несколько раздельных массивов) и оптимизированная структура (чередование данных массивов в памяти для улучшения пространственной локальности). Настраиваемый параметр — количество одновременно обрабатываемых массивов (ось X), измеряемая характеристика — время выполнения/число тактов (ось Y). Графики представлены на рисунке 7: красная кривая — неоптимизированный вариант, зелёная — оптимизированный.

Результат эксперимента: при росте числа массивов неоптимизированный вариант показывает заметное увеличение времени (ухудшение из-за слабой локальности и неэффективного использования кэш-линий), тогда как оптимизированный вариант сохраняет значительно более низкое время выполнения.

Вывод: переструктурирование данных под характер доступа повышает долю полезных данных в каждой кэш-линии и уменьшает потери на обращениях к памяти.

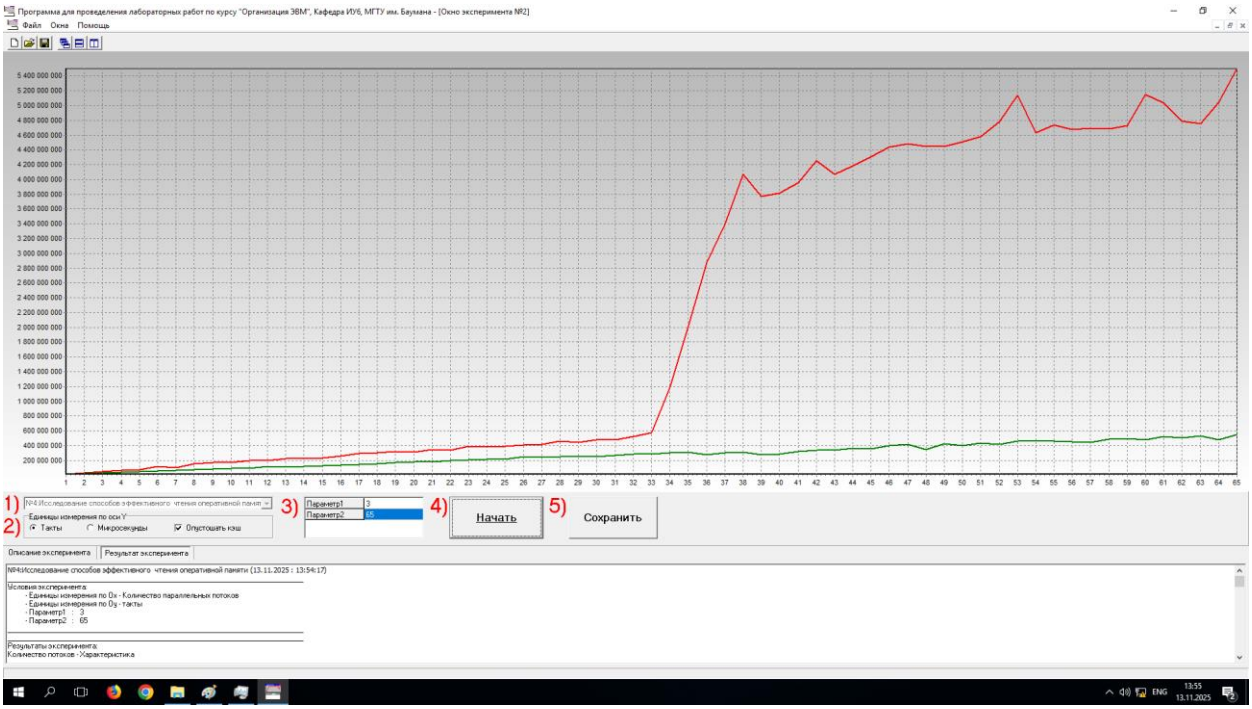

Рисунок 7 – Результат 4ого эксперимента

На рисунке 8 приведены описания параметров эксперимента.

Параметр	Диапазон	Масштаб	Описание
№ 1	1..4	M	Размер массива
№ 2	1..128	1	Количество потоков данных

Рисунок 8 – Описание параметров 4ого эксперимента

Эксперимент 5 «*Исследование конфликтов в кеш-памяти*»

Исходные данные эксперимента: сравнивались две процедуры чтения и обработки данных — процедура, намеренно создающая конфликты в кэше (чтение с шагом, кратным размеру банка/набора), и процедура, избегающая конфликтов за счёт выбора смещения читаемой ячейки на шаг, соответствующий размеру кэш-линейки. Настраиваемый параметр — смещение читаемой ячейки от начала блока (ось X), измеряемая характеристика — время выполнения/число тактов (ось Y). Результаты приведены на рисунке 9: красная кривая — с конфликтами, зелёная — без конфликтов.

Результат эксперимента: вариант с конфликтами демонстрирует существенно большее время выполнения, в то время как устранение конфликтов за счёт смещения резко снижает затраты.

Вывод: конфликтные обращения приводят к вытеснениям и росту промахов кэша, поэтому корректный выбор шага/смещения доступа является практическим способом оптимизации производительности без изменения вычислений.

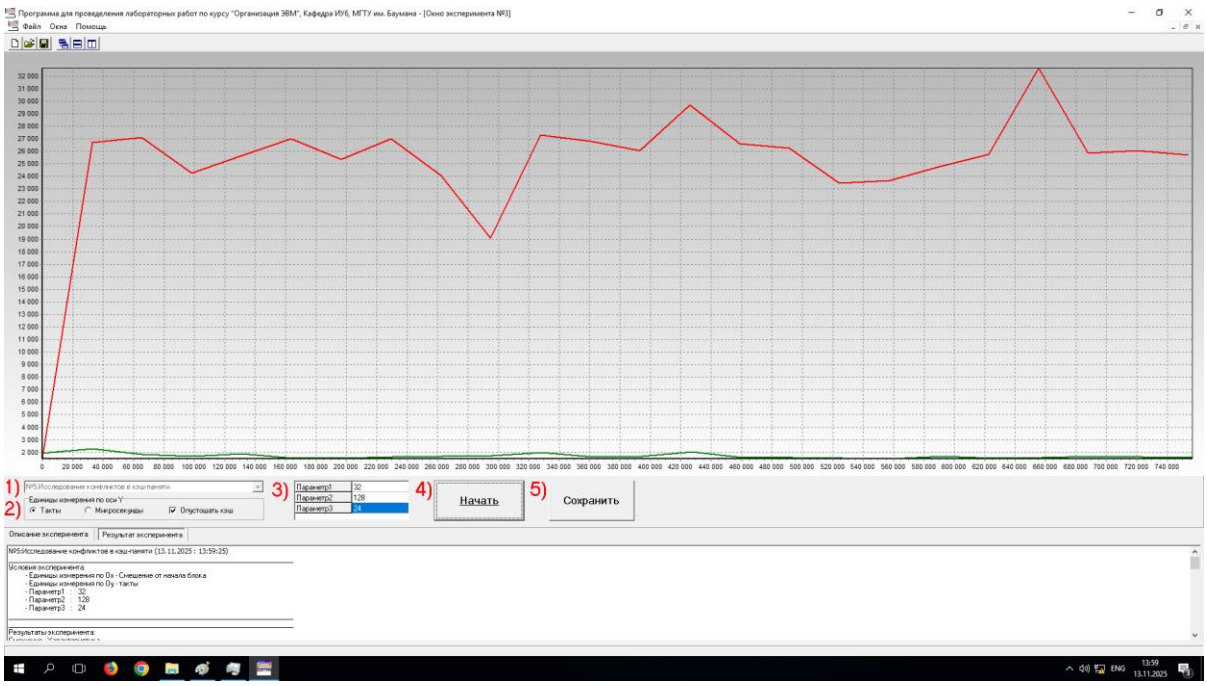

Рисунок 9 – Результат 5ого эксперимента

На рисунке 10 приведены описания параметров эксперимента.

Параметр	Диапазон	Масштаб	Описание
№ 1	1..256	К	Размер банка кэш-памяти
№ 2	1..128	б	Размер линейки кэш-памяти
№ 3	2..512	1	Количество читаемых линеек

Рисунок 10 – Описание параметров 5ого эксперимента

Эксперимент 6 «*Сравнение алгоритмов сортировки*»

Исходные данные эксперимента: сравнивались QuickSort, неоптимизированный Radix-Counting и оптимизированный под 8-процессорную систему Radix-Counting. Настраиваемый параметр — количество 64-разрядных элементов сортируемого массива (ось X), измеряемая характеристика — время выполнения/число тактов (ось Y). На рисунке 11 фиолетовая линия соответствует QuickSort, красная — неоптимизированному Radix-Counting, зелёная — оптимизированному Radix-Counting.

Результат эксперимента: по мере роста размера массива различия между алгоритмами увеличиваются; оптимизированный Radix-Counting демонстрирует наименьшее время выполнения, а QuickSort — наибольшее среди представленных кривых.

Вывод: алгоритмы с линейной/квазилинейной трудоёмкостью и распараллеливанием выигрывают на больших объёмах данных, тогда как сравнимые по смыслу «неоптимизированные» варианты заметно уступают из-за худшего использования вычислительных ресурсов и памяти.

Рисунок 11 – Результат 6ого эксперимента

На рисунке 11 приведены описания параметров эксперимента.

№ 1	1..20	M	Количество 64-х разрядных элементов массивов
№ 2	4..1024	K	Шаг увеличения размера массива

Рисунок 10 – Описание параметров 6ого эксперимента

Вывод

В ходе лабораторной работы были экспериментально исследованы особенности работы подсистемы памяти и кэширования на реальной вычислительной системе: по временным характеристикам доступа оценены параметры организации динамической памяти (расслоение/страничность), выполнено сравнение ссылочных и векторных структур данных и показано влияние локальности и фрагментации на производительность. Также изучены методы повышения эффективности чтения ОЗУ — программная предвыборка, переструктурирование данных и выбор смещения для устранения конфликтов в кэше. Дополнительно проведено сравнение алгоритмов сортировки, продемонстрировавшее преимущество оптимизированного и

распараллеленного Radix-Counting на больших объёмах данных по сравнению с QuickSort и неоптимизированной реализацией.