|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**ОТЧЕТ**

**по лабораторной работе № 2**

**Название:** Организация ЭВМ и систем

**Дисциплина:** Организация памяти конвейерных суперскалярных

электронных вычислительных машин

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-72Б |  |  | И.С. Марчук |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | А.Ю. Попов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Цель работы** - освоение принципов эффективного использования подсистемы памяти современных универсальных ЭВМ, обеспечивающей хранение и своевременную выдачу команд и данных в центральное процессорное устройство.

**Ход работы**

**Эксперимент 1.** Исследования расслоения динамической памяти.

**Цель:** определение способа трансляции физического адреса, используемого при обращении к динамической памяти.

Смоделированный эксперимент показан на рисунке 1.

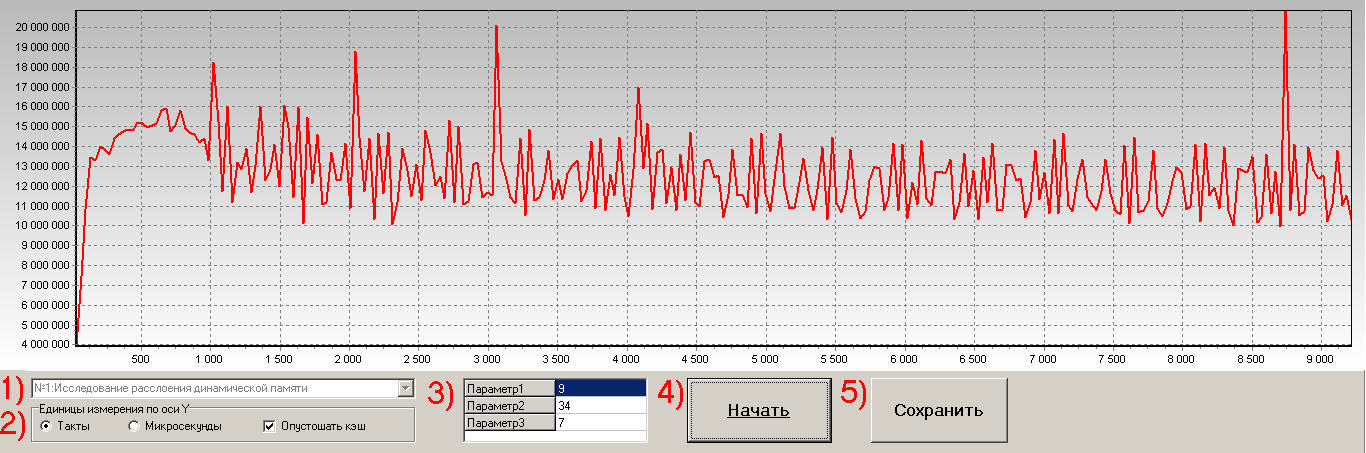


Рисунок 1

**Вывод:** Переход к близким данным переходится быстрее. Данные лучше сохранить в один пакет. Надо чтобы было чередование.

**Эксперимент 2.** Сравнение эффективности ссылочных и векторных структур.

**Цель:** оценка влияния зависимости команд по данным на эффективность вычислений.

**Исходные данные:** размер линейки кэш-памяти верхнего уровня; объем физической памяти.

Смоделированный эксперимент показан на рисунке 2.

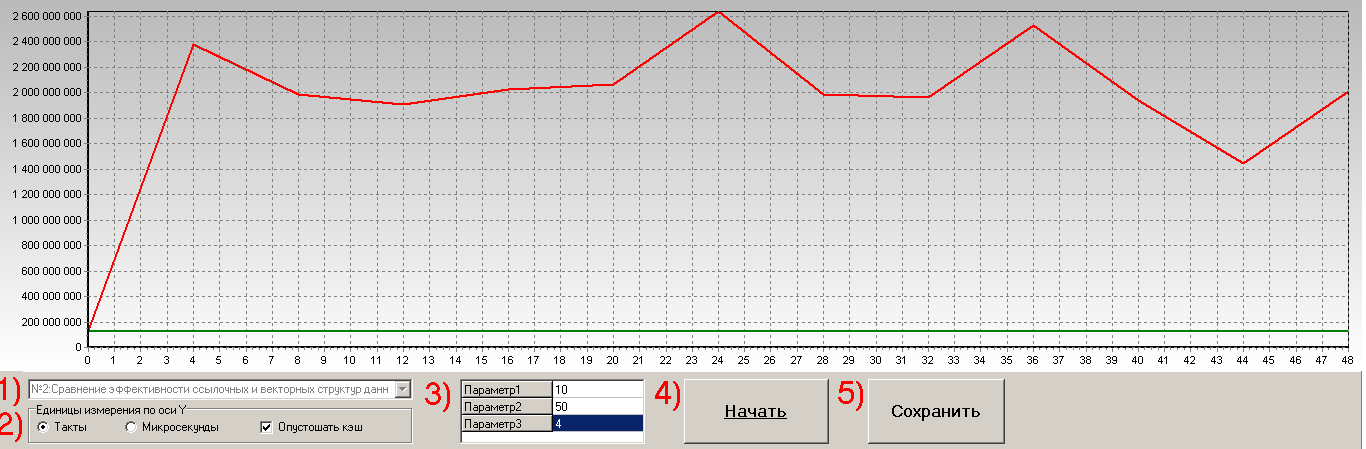


Рисунок 2

**Вывод:** Массив — примитивная векторная структура, но их можно везде применить. Но иногда нужны списки. Надо улучшать ЭВМ, чтобы решить проблему семантического разрыва.

**Эксперимент 3.** Исследование эффективности программной предвыборки.

**Цель:** выявление способов ускорения вычислений благодаря применению предвыборки данных.

**Исходные данные:** степень ассоциативности и размер TLB данных.

Смоделированный эксперимент показан на рисунке 3.

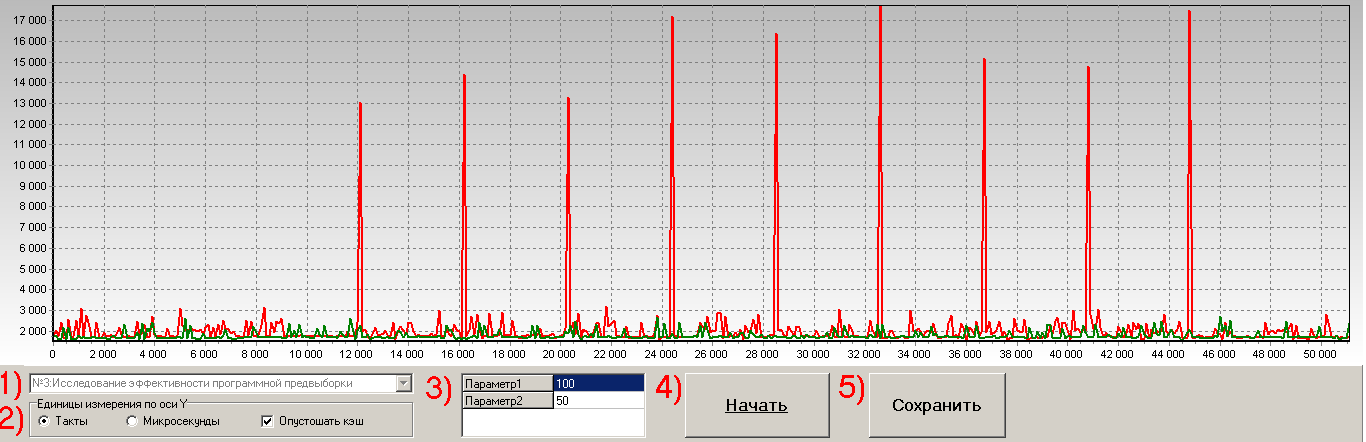


Рисунок 3

**Вывод:** можно использовать предварительную загрузку в TLB (если не будет прерывания) в ограниченное время. Что сделает обращение к памяти в 1,3 раза быстрее.

**Эксперимент 4.** Исследование способов эффективного чтения оперативной памяти.

**Цель:** исследование возможности ускорения вычислений благодаря использованию структур данных, оптимизирующих механизм чтения оперативной памяти.

**Исходные данные**: Адресное расстояние между банками памяти, размер буфера чтения.

Смоделированный эксперимент показан на рисунке 4.

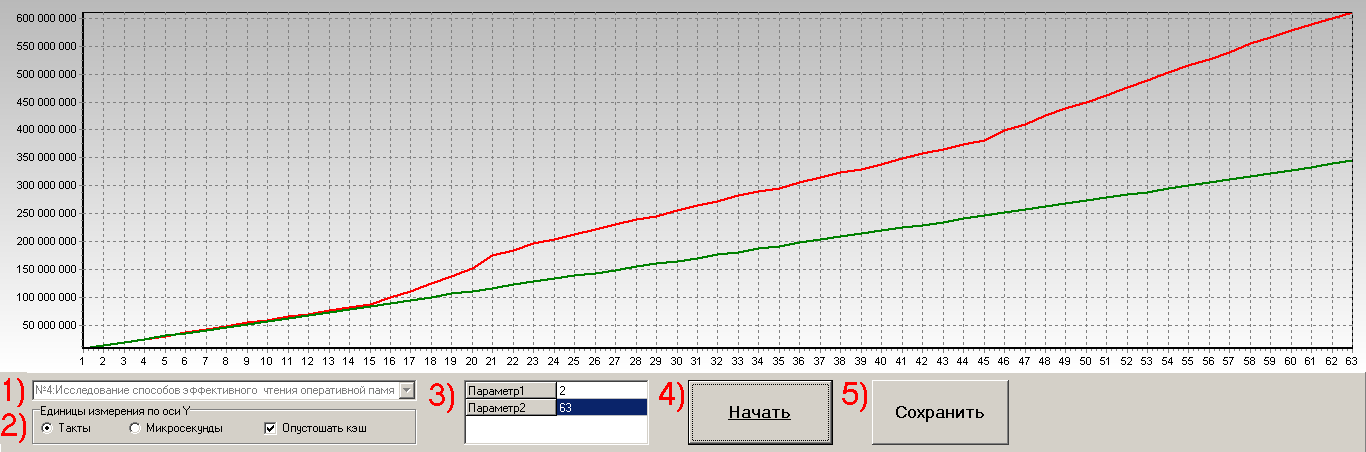


Рисунок 4

Вывод: при оптимизации необходимо, чтобы данные были расположены в одном пакете, что даст быстродействие. Если это возможно реализовать в программировании, то это дат ускорение.

**Эксперимент 5.** Исследование конфликтов в кэш-памяти.

**Цель:** исследование влияния конфликтов кэш-памяти на эффективность вычислений.

**Исходные данные:** Размер банка кэш-памяти данных первого и второго уровня, степень ассоциативности кэш-памяти первого и второго уровня, размер линейки кэш памяти первого и второго уровня.

Смоделированный эксперимент показан на рисунке 5.

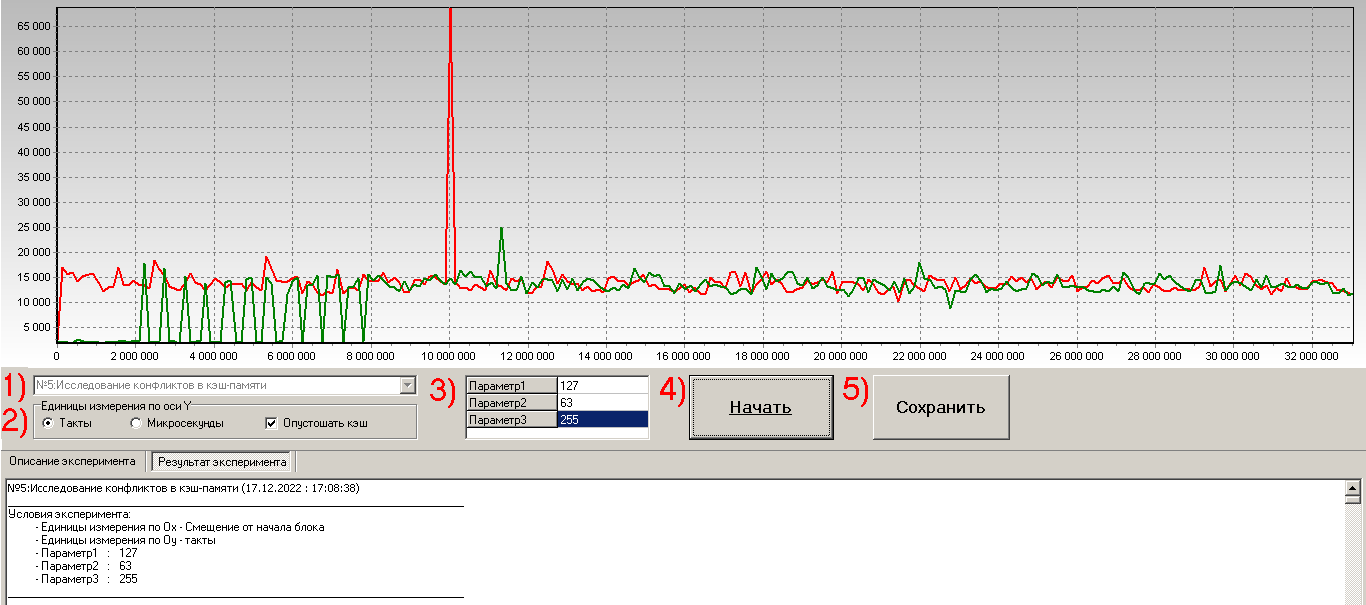


Рисунок 5

**Вывод:** нужно “ходить” по памяти, не равный расстоянию конфликта. С кэшем в 6 раз быстрее. Надо уменьшать конфликты в кэш памяти.

**Эксперимент 6.** Сравнение алгоритмов сортировки.

**Цель:** исследование способов эффективного использования памяти и выявление наиболее эффективных алгоритмов сортировки, применимых в вычислительных системах.

**Исходные данные:** количество процессоров вычислительной системы, размер пакета, количество элементов в массиве, разрядность элементов массива.

Смоделированный эксперимент показан на рисунке 6.

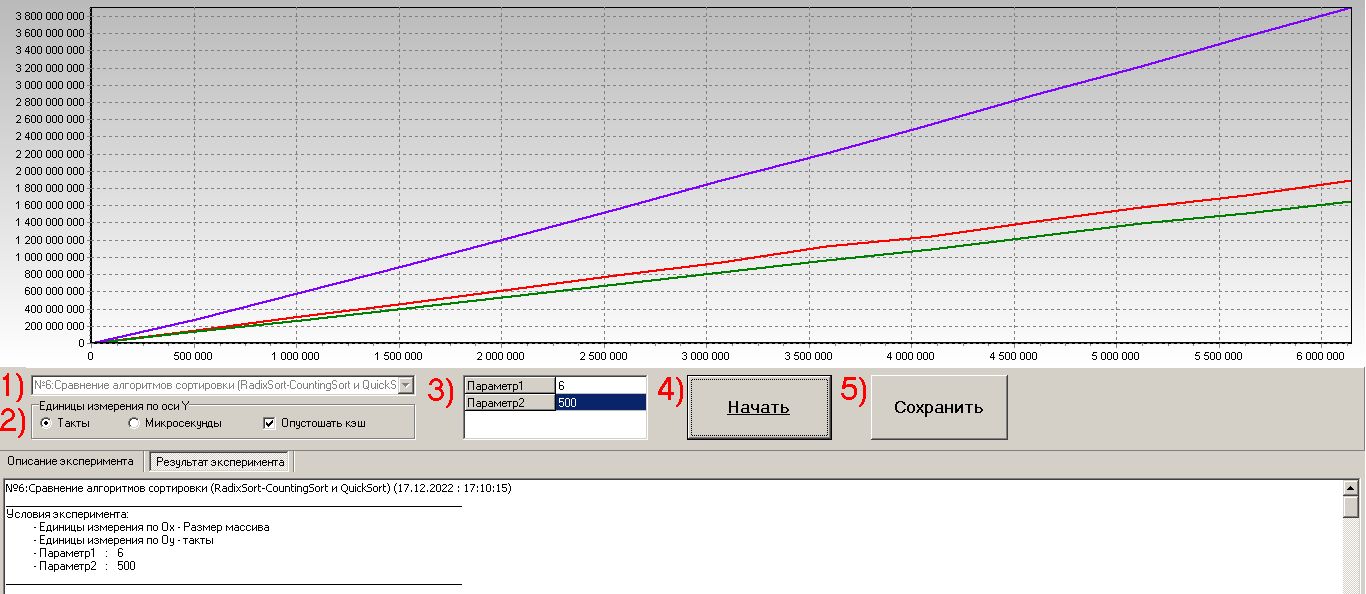


Рисунок 6

**Вывод:** несмотря на формульную быстроту QuickSort – в среденем O(n\*log(n)) и O(n2), у метода Radix-Counting формульная быстрота работы будет О(n/log(n)), что говорит о его быстродействии.

**Вывод:** узнали принципы эффективного использования подсистемы памяти современных универсальных ЭВМ, обеспечивающее хранение и своевременную выдачу команд и данных в центральное процессорное устройство. Изучили программу PCLAB.