



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»

## Лабораторная работа №3 по дисциплине «Архитектура ЭВМ»

Тема Организация памяти конвейерных суперскалярных электронных вычислительных машин

Студент Тузов Даниил Александрович

Группа ИУ7-52Б

Преподаватель Калитвенец Максим, Попов А.Ю.

Москва, 2024 г.

# 1 Введение

**Цель работы** – освоение принципов эффективного использования подсистемы памяти современных универсальных ЭВМ, обеспечивающей хранение и своевременную выдачу команд и данных в центральное процессорное устройство. Для достижения поставленной цели необходимо решить следующие задачи:

- 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 об архитектурных особенностях используемых ЭВМ.

Все замеры проводились на ЭВМ, характеристики которой приведены ниже:

- процессор – 12th Gen Intel(R) Core(TM) i5-12450H 2.00 ГГц;
- оперативная память – 16,0 ГБ;
- тип системы – 64-разрядная операционная система, процессор x64;
- операционная система – Windows 11;
- версия ОС – 23H2.

## 2 Эксперимент 1. Исследование расслоения динамической памяти

### 2.1 Условия эксперимента

Условия эксперимента:

- Единицы измерения по Ох - Байты
- Единицы измерения по Оу - такты
- Параметр1 : 16
- Параметр2 : 16
- Параметр3 : 16

Рисунок 1 – Условия эксперимента 1

### 2.2 Результат эксперимента

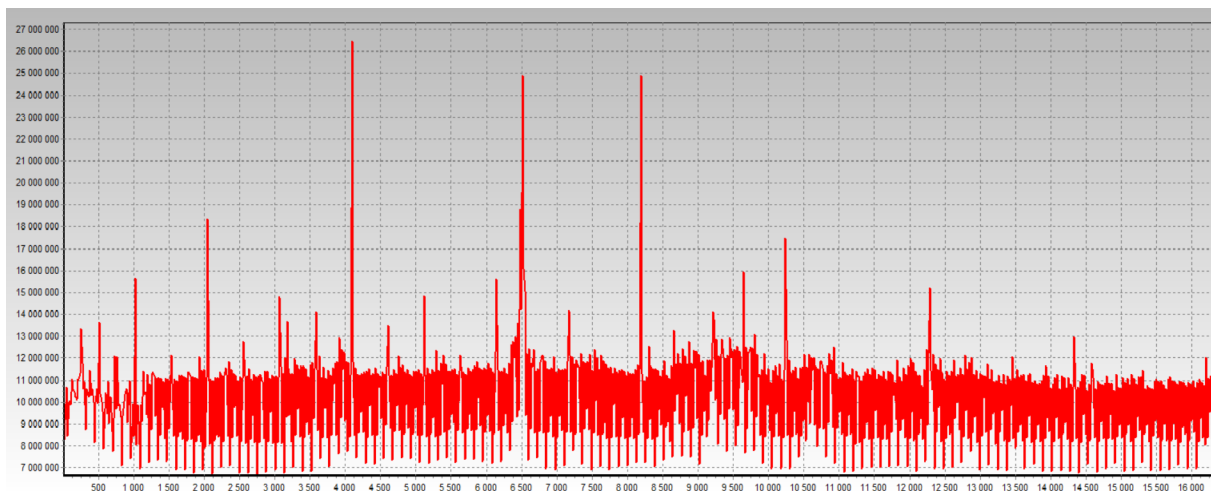


Рисунок 2 – Результат эксперимента 1

### 2.3 Вывод

Исходя из графика, минимальный шаг, при котором происходит постоянное обращение к одному и тому же банку  $T1 = 128$ ; объем данных, являющийся минимальной порцией обмена кэш-памяти верхнего уровня с оперативной памятью  $\Pi = 8$ ; количество банков памяти  $B = 16$ ; шаг чтения являющийся наихудшим при обращении к динамической памяти  $T2 = 4096$ .

## 3 Эксперимент 2. Сравнение эффективности ссылочных и векторных структур

### 3.1 Условия эксперимента

Условия эксперимента:  
- Единицы измерения по Ох - Килобайты  
- Единицы измерения по Оу - такты  
- Параметр1 : 5  
- Параметр2 : 50  
- Параметр3 : 2

Рисунок 3 – Условия эксперимента 2

### 3.2 Результат эксперимента

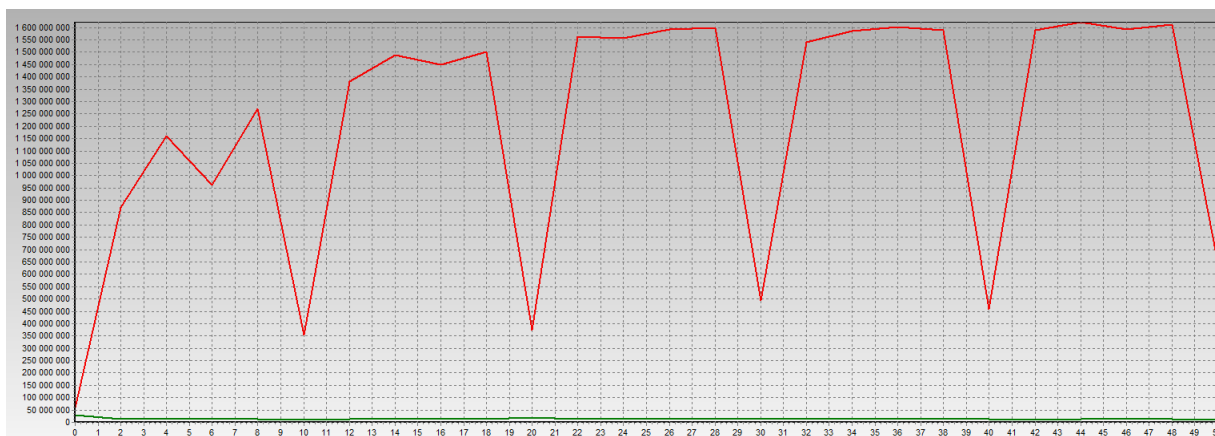


Рисунок 4 – Результат эксперимента 2

### 3.3 Вывод

Красный график показывает время или количество тактов работы алгоритма, использующего список. Зеленый график показывает время или количество тактов работы алгоритма, использующего массив. Очевидно, массив быстрее.

## 4 Эксперимент 3. Исследование эффективности программной предвыборки

### 4.1 Условия эксперимента

Условия эксперимента:  
- Единицы измерения по Ох - Байты  
- Единицы измерения по Оу - такты  
- Параметр1 : 512  
- Параметр2 : 1024

Рисунок 5 – Условия эксперимента 3

### 4.2 Результат эксперимента

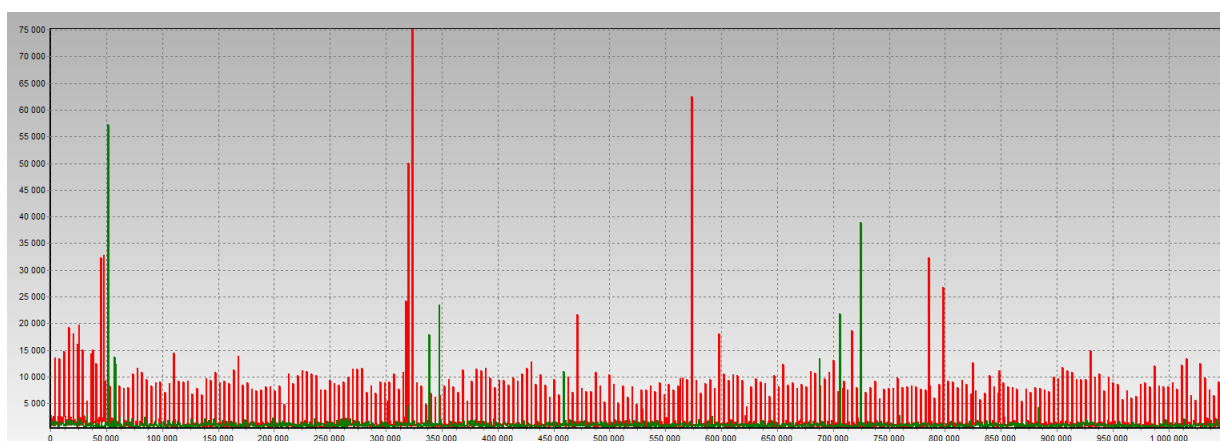


Рисунок 6 – Результат эксперимента 3

### 4.3 Вывод

Красный график показывает время или количество тактов работы алгоритма без предвыборки. Зеленый график показывает время или количество тактов работы алгоритма с использованием предвыборки. Алгоритм с использованием предвыборки работает быстрее.

## 5 Эксперимент 4. Исследование способов эффективного чтения оперативной памяти

### 5.1 Условия эксперимента

Условия эксперимента:  
- Единицы измерения по Ох - Количество параллельных потоков  
- Единицы измерения по Оу - такты  
- Параметр1 : 4  
- Параметр2 : 70

Рисунок 7 – Условия эксперимента 4

### 5.2 Результат эксперимента

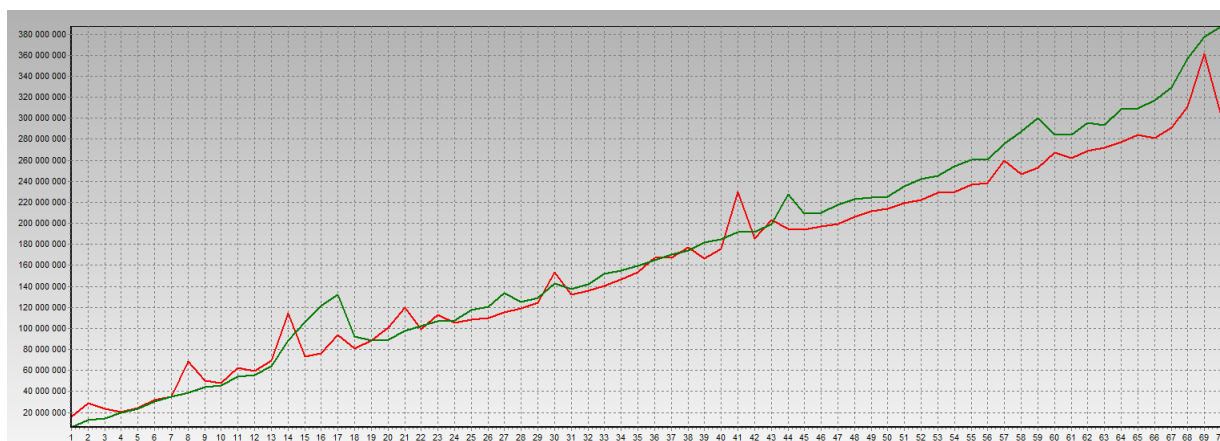


Рисунок 8 – Результат эксперимента 4

### 5.3 Вывод

Красный график показывает время или количество тактов работы алгоритма, использующего неоптимизированную структуру. Зеленый график показывает время (или количество тактов) работы алгоритма с использованием оптимизированной структуры. Алгоритм, использующий неоптимизированную структуру, быстрее.

## 6 Эксперимент 5. Исследование конфликтов в кэш-памяти

### 6.1 Условия эксперимента

Условия эксперимента:  
- Единицы измерения по Ох - Смещение от начала блока  
- Единицы измерения по Оу - такты  
- Параметр1 : 64  
- Параметр2 : 16  
- Параметр3 : 64

Рисунок 9 – Условия эксперимента 5

### 6.2 Результат эксперимента

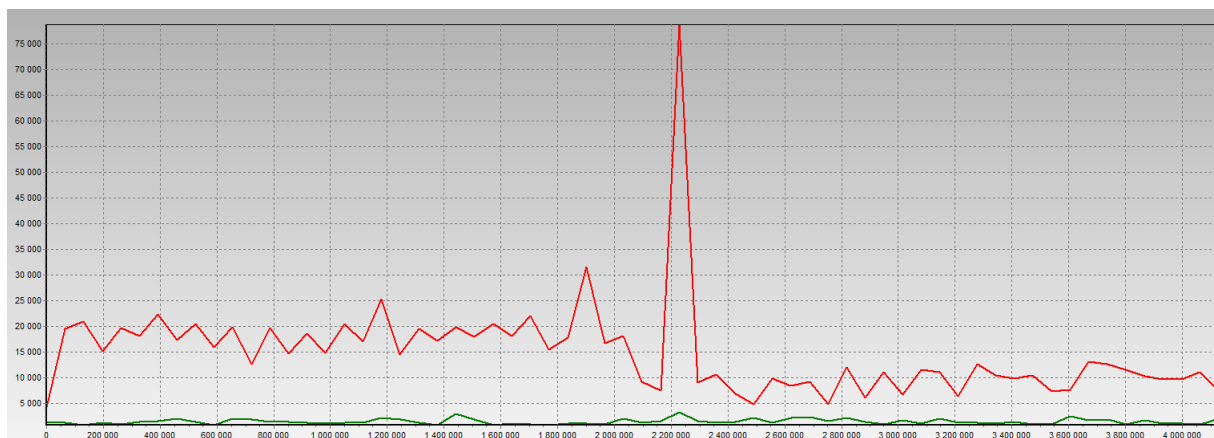


Рисунок 10 – Результат эксперимента 5

### 6.3 Вывод

Красный график показывает время или количество тактов работы процедуры, читающей данные с конфликтами в кэш-памяти. Зеленый график показывает время или количество тактов работы процедуры, не вызывающей конфликтов в кэш-памяти. Процедура, не вызывающая конфликтов в кэш-памяти, быстрее.

## 7 Эксперимент 6. Сравнение алгоритмов сортировки

### 7.1 Условия эксперимента

Условия эксперимента:  
- Единицы измерения по Ох - Размер массива  
- Единицы измерения по Оу - такты  
- Параметр1 : 10  
- Параметр2 : 512

Рисунок 11 – Условия эксперимента 6

### 7.2 Результат эксперимента

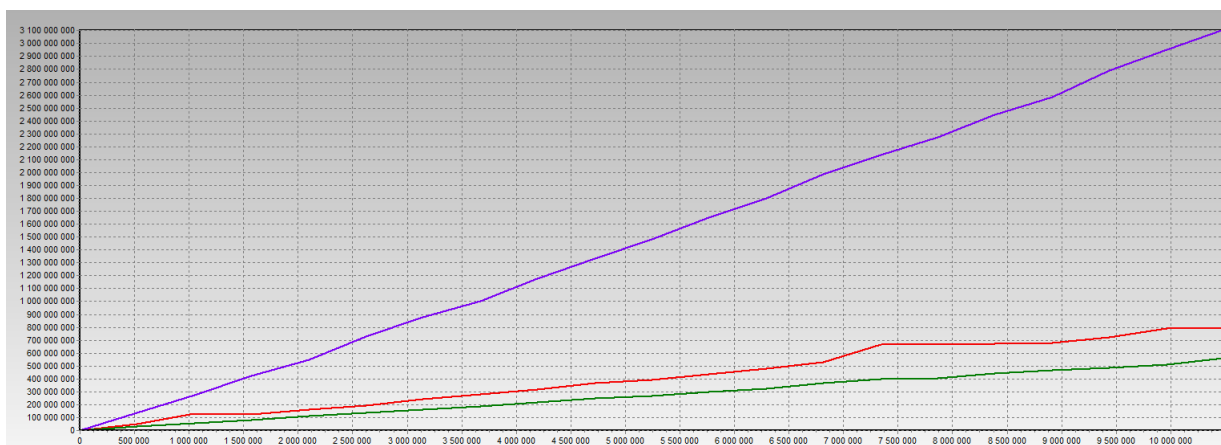


Рисунок 12 – Результат эксперимента 6

### 7.3 Вывод

Фиолетовый график показывает время или количество тактов работы алгоритма QuickSort. Красный график показывает время или количество тактов работы неоптимизированного алгоритма Radix-Counting. Зеленый график показывает время или количество тактов работы оптимизированного под 8-процессорную вычислительную систему алгоритма Radix-Counting. Самый быстрый алгоритм – оптимизированный Radix-Counting, самый медленный – QuickSort.



## 8 Ответы на контрольные вопросы

1. **Вопрос:** назовите причины расслоения оперативной памяти. **Ответ:** в связи с конструктивной неоднородностью оперативной памяти, обращение к последовательно расположенным данным требует различного времени.
2. **Вопрос:** как в современных процессорах реализована аппаратная предвыборка. **Ответ:** с помощью кэша или специальных таблиц адресов переходов BTB (Branch Target Buffer).
3. **Вопрос:** какая информация храниться в TLB. **Ответ:** TLB хранит последние переводы виртуальной памяти в физическую память и может называться кэшем преобразования адресов.
4. **Вопрос:** какой тип ассоциативной памяти используется в кэш-памяти второго уровня современных ЭВМ и почему. **Ответ:**
5. **Вопрос:** приведите пример программной предвыборки. **Ответ:** получение информации о преобразовании адресов из TLB буфера.