|  |  |
| --- | --- |
|  | **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 памяти конвейерных суперскалярных ЭВМ

**Дисциплина:** Организация ЭВМ и систем

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-72Б |  |  | С.В. Астахов | |
|  | (Группа) |  | (Подпись, дата) | | (И.О. Фамилия) |
|  |  |  |  | |  |
| Преподаватель |  |  |  | |  |
|  |  |  | (Подпись, дата) | | (И.О. Фамилия) |

Москва, 2022

**Введение**

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

**Задание 1**

Ознакомиться с возможностями программы PCLAB в Разделе 2 методических указаний. Запустить программу PCLAB 1.0. Изучить идентификационную информацию на вкладке «Идентификация процессора».

Идентификационная информация приведена в приложении А.

**Задание 2**

На основании идентификационной информации о микропроцессоре ЭВМ, используемой при проведении лабораторной работы, определить следующие параметры: размер линейки кэш-памяти верхнего уровня и объем физической памяти. Результаты занести в отчет.

Длина строки —128 Б

Размер — 1 МБ

**Задание 3**

Ознакомиться с описанием эксперимента «Исследование расслоения динамической памяти» на вкладке «Описание эксперимента». Провести эксперимент. По результатам эксперимента определить: количество банков динамической памяти; размер одной страницы динамической памяти; количество страниц в динамической памяти. Сделать выводы о использованном способе наращивания динамической памяти. Результаты занести в отчет.

Условия эксперимента:

                  - Единицы измерения по Ох - Байты

                  - Единицы измерения по Оу - такты

                  - Параметр1      :      32

                  - Параметр2      :      128

                  - Параметр3      :      1

Результаты проведенного эксперимента показаны на рисунке 1.

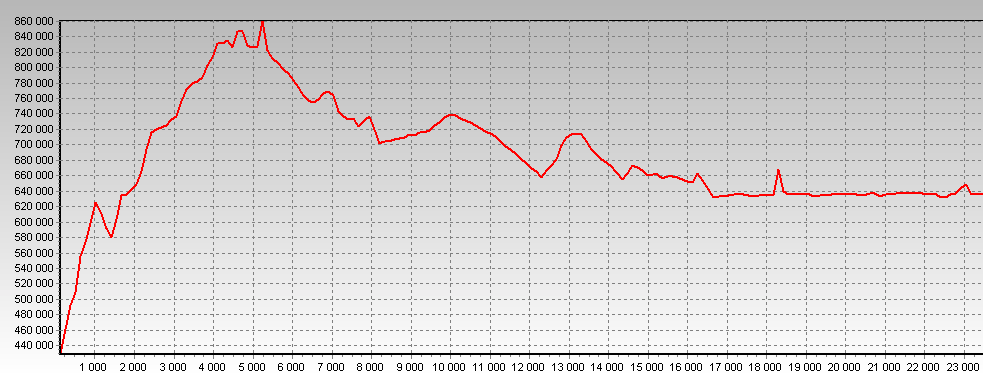


Рисунок 1 — Исследование расслоения динамической памяти

Т1 = 1024 — 1ый экстремум функции

П = 128 — размер линейки кэш-памяти

Б = Т1 / П = 1024 / 128 = 8 — число банков

Т2 = 4096 — глобальный экстремум функции

PC = T2 / Б = 4096 / 8 = 512 — размер одного банка

О = 64 К — полный объем памяти

С = О / (РС\*Б\*П) = 2^20 / (512\*8\*128) = 2^(20-9-3-7) = 2

Вывод: В связи с конструктивной неоднородностью оперативной памяти, обращение к последовательно расположенным данным требует различного времени. Переход к близким данным осуществляется быстрее. Необходимо чередовать запросы к банкам памяти.

**Задание 4**

Ознакомиться с описанием эксперимента «Сравнение эффективности ссылочных и векторных структур данных». Провести эксперимент. По результатам эксперимента определить: отношение времени работы алгоритма, использующего зависимые данные, ко времени обработки аналогичного алгоритма обработки независимых данных. Сделать выводы об эффективности ссылочных и векторных структур данных и способах ее повышения. Результаты занести в отчет.

Условия эксперимента:

                  - Единицы измерения по Ох - Килобайты

                  - Единицы измерения по Оу - такты

                  - Параметр1      :      1

                  - Параметр2      :      32

                  - Параметр3      :      1

Результаты эксперимента представлены на рисунке 2.

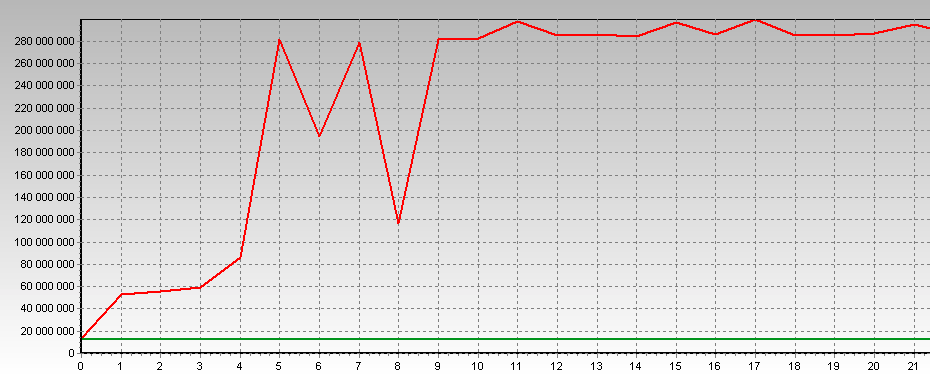


Рисунок 2 — Сравнение эффективности ссылочных и векторных структур данных

Список обрабатывался в 19,645843 раз дольше.

Вывод: по результатам эксперимента видно, что ссылочные структуры (особенно, с ростом фрагментации) требуют большего времени для обработки, чем векторные, так как в случае ссылочных структур происходит обработка зависимых данных, т.е. адрес загружаемого операнда становится известным только после обработки предыдущей команды.

**Задание 5**

Для ЭВМ, используемой при проведении лабораторной работы определить следующие параметры: степень ассоциативности и размер TLB данных. Ознакомиться с описанием и провести эксперимент «Исследование эффективности программной предвыборки». По результатам эксперимента определить: отношение времени последовательной обработки блока данных ко времени обработки блока с применением предвыборки; время и количество тактов первого обращения к странице данных. Сделать выводы об эффективности предвыборки и способах ее повышения. Результаты занести в отчет.

Условия эксперимента:

                  - Единицы измерения по Ох - Байты

                  - Единицы измерения по Оу - такты

                  - Параметр1      :      512

                  - Параметр2      :      64

Результаты эксперимента представлены на рисунке 3.

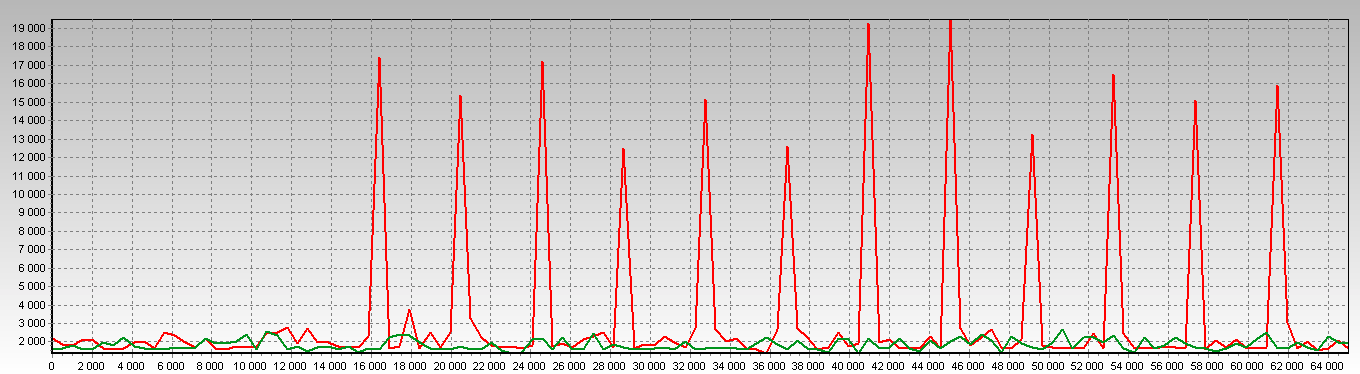


Рисунок 3 — Исследование эффективности программной предвыборки

Обработка без загрузки таблицы страниц в TLB производилась в 1,783955 раз дольше.

Число тактов первого обращения к странице данных — 17412.

Вывод: программная предвыборка позволяет сократить время доступа к оперативной памяти и стабилизировать задержку, исключая двойное обращение к оперативной памяти.

**Задание 6**

Ознакомиться с описанием и провести эксперимент «Исследование способов эффективного чтения оперативной памяти». По результатам эксперимента определить: отношение времени обработки блока памяти неоптимизированной структуры ко времени обработки блока структуры, обеспечивающей эффективную загрузку и параллельную обработку данных. Сделать выводы о способах повышения эффективности чтения оперативной памяти.

Условия эксперимента:

                  - Единицы измерения по Ох - Количество параллельных потоков

                  - Единицы измерения по Оу - такты

                  - Параметр1      :      2

                  - Параметр2      :      64

Результаты эксперимента представлены на рисунке 4.

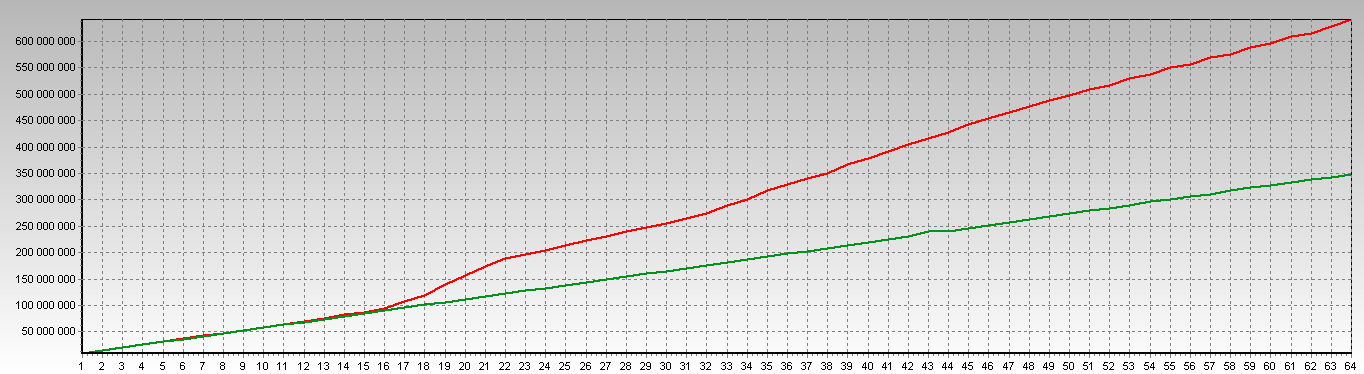


Рисунок 4 — Исследование способов эффективного    чтения оперативной памяти

Неоптимизированная структура обрабатывалась в 1,6762589 раз дольше.

Вывод: при использовании неоптимизированных структур данных они распределяются по пакетам некомпактно, что вызывает задержки в работе программы. Оптимизация структур данных, например, более «плотное» распределение данных по пакетам, позволяет ускорить программу.

**Задание 7**

Для ЭВМ, используемой при проведении лабораторной работы определить следующие параметры: 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. Ознакомиться с описанием и провести эксперимент «Исследование конфликтов в кэш-памяти». По результатам эксперимента определить: отношение времени обработки массива с конфликтами в кэш-памяти ко времени обработки массива без конфликтов. Сделать выводы о способах устранения конфликтов в кэш-памяти.

L1 размер банка - 1 МБ

L2 размер банка - 8 МБ

Обе 8-ассоциативны

Размеры страницы - 128Б и 1КБ соответственно

Условия эксперимента:

                  - Единицы измерения по Ох - Смещение от начала блока

                  - Единицы измерения по Оу - такты

                  - Параметр1      :      128

                  - Параметр2      :      128

                  - Параметр3      :      32

Результаты эксперимента представлены на рисунке 5.

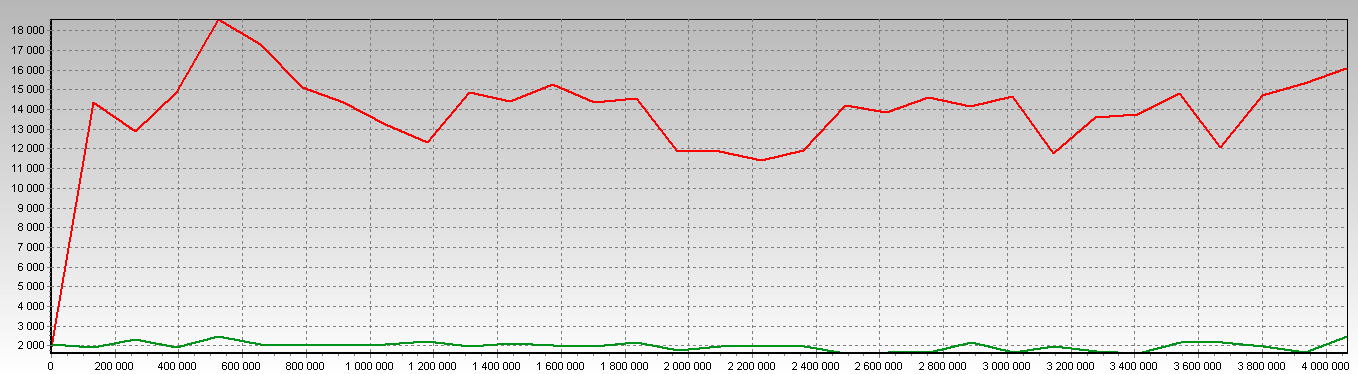


Рисунок 5 — Исследование конфликтов в кэш-памяти

Чтение с конфликтами банков производилось в 6,9114675 раз дольше.

Вывод: программа без конфликтов кеш-памяти (кеширующая данные не в один и тот же набор) работает значительно быстрее. Для оптимизации программы следует обращаться к памяти на расстоянии не кратном размеру банка.

**Задание 8**

Ознакомиться с описанием и провести эксперимент «Исследование алгоритмов сортировки». По результатам эксперимента определить: отношение времени сортировки массивов алгоритмом QuickSort ко времени сортировки алгоритмом Counting- Radix, а также ко времени сортировки Counting-Radix алгоритмом, оптимизированным под 8-процессорную вычислительную систему. Сделать выводы о наиболее эффективном алгоритме сортировки.

Условия эксперимента:

                  - Единицы измерения по Ох - Размер массива

                  - Единицы измерения по Оу - такты

                  - Параметр1      :      1

                  - Параметр2      :      128

Результаты эксперимента приведены на рисунке 6.

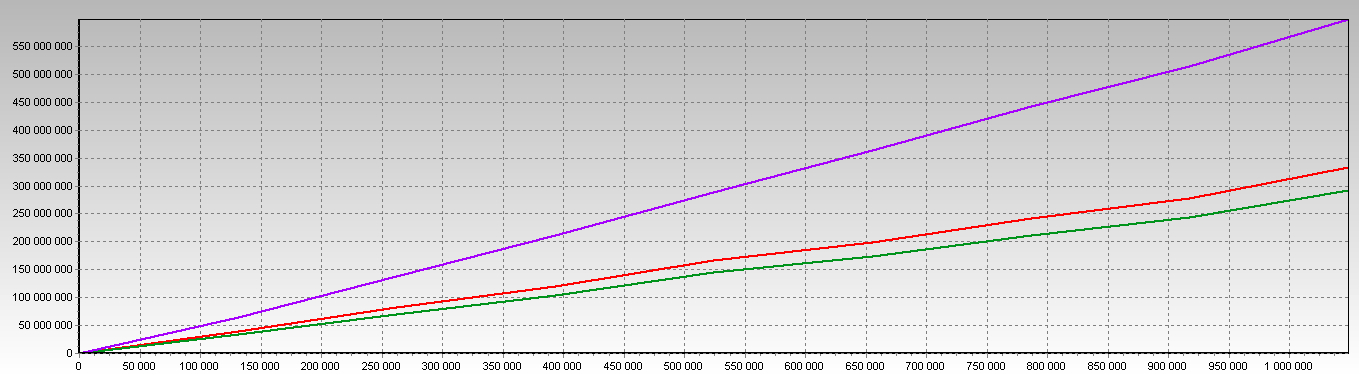


Рисунок 6 — Исследование алгоритмов сортировки

QuickSort работал в 1,7959224 раз дольше Radix-Counting Sort.

QuickSort работал в 2,0576808 раз дольше Radix-Counting Sort, оптимизированного под 8-процессорную ЭВМ.

Вывод: метод Radix-Counting имеет меньшую вычислительную сложность, чем QuickSort, так как работает без использования парных сравнений, однако, вычислительная сложность алгоритма существенно зависит от его настройки на определенную разрядность чисел и размерность массивов.

**Вывод:** результатом выполнения данной работы является освоение принципов эффективного использования подсистемы памяти современных универсальных ЭВМ, обеспечивающей хранение и своевременную выдачу команд и данных в центральное процессорное устройство.

**ПРИЛОЖЕНИЕ А**

**Информация о процессоре**

eax in eax ebx ecx edx

00000000 0000000d 756e6547 6c65746e 49656e69

00000001 0001067a 00020800 0400e3bd bfebfbff

00000002 05b0b101 005657f0 00000000 2cb43078

00000003 00000000 00000000 00000000 00000000

00000004 00000000 00000000 00000000 00000000

00000005 00000040 00000040 00000003 00022220

00000006 00000001 00000002 00000003 00000000

00000007 00000000 00000000 00000000 00000000

00000008 00000400 00000000 00000000 00000000

00000009 00000000 00000000 00000000 00000000

0000000a 07280202 00000000 00000000 00000503

0000000b 00000000 00000000 00000000 00000000

0000000c 00000000 00000000 00000000 00000000

0000000d 00000000 00000000 00000000 00000000

80000000 80000008 00000000 00000000 00000000

80000001 00000000 00000000 00000001 20000000

80000002 65746e49 2952286c 6c654320 6e6f7265

80000003 20295228 20555043 20202020 45202020

80000004 30303333 20402020 30352e32 007a4847

80000005 00000000 00000000 00000000 00000000

80000006 00000000 00000000 04004040 00000000

80000007 00000000 00000000 00000000 00000000

80000008 00003024 00000000 00000000 00000000

Undocument layers

80860000 00000000 00000000 00000000 00000000

80860001 00000000 00000000 00000000 00000000

80860002 00000000 00000000 00000000 00000000

80860003 00000000 00000000 00000000 00000000

80860004 00000000 00000000 00000000 00000000

80860005 00000000 00000000 00000000 00000000

80860006 00000000 00000000 00000000 00000000

80860007 00000000 00000000 00000000 00000000

c0000000 00000000 00000000 00000000 00000000

c0000001 00000000 00000000 00000000 00000000

8ffffffe 00000000 00000000 00000000 00000000

8fffffff 00000000 00000000 00000000 00000000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Vendor ID: "GenuineIntel"; CPUID level 13

Дополнительные функции Intel:

Верисия 0001067a:

Type 0 - Original OEM

Family 6 - Pentium Pro

Model 7 - Pentium III/Pentium III Xeon - external L2 cache

Stepping 10

Reserved 4

Extended brand string: "Intel(R) Celeron(R) CPU E3300 @ 2.50GHz"

CLFLUSH instruction cache line size: 8

Hyper threading siblings: 2

Feature flags bfebfbff:

0 FPU Присутствует Математический сопроцессор

1 VME Поддержка расширенных возможностей обработки прерываний в режиме виртуального i8086

2 DE Поддержка отладки

3 PSE Поддержка страниц размером 4 MB

4 TSC Счетчик меток реального времени

5 MSR Поддержка команд rdmsr и wrmsr

6 PAE Поддержка физического адреса более 32 бит

7 MCE Поддержка исключений 18 - об аппаратных ошибках

8 CX8 Поддержка инструкции cmpxchg8b

9 APIC Микропроцессор содержит программно доступный контроллер прерываний

11 SEP Поддержка инструкций быстрых системных вызовов sysenter и sysexit

12 MTRR Поддержка регистра mtrr\_cap (относится к MSR-регистрам)

13 PGE Поддержка глобальных страниц

14 MCA Поддержка архитектуры машинного контроля

15 CMOV Поддержка инструкций условной пересылки cmov, fcmovcc, fcomi

16 PAT Процессор поддерживает таблицу атрибутов страницы

17 PSE-36 Процессор поддерживает 4 MB страницы, которые способны адресовать физическую память до 64 GB

19 CLFLSH Поддержка инструкции CLFLUSH

21 DS Поддержка записи отладочной информации

22 ACPI Управление охлаждением процессора с помощью пустых циклов в зависимости от температуры

23 MMX Поддержка MMX

24 FXSR Поддержка инструкций FXSAVE и FXRSTOR

25 SSE Поддержка SSE

26 SSE2 Поддержка SSE2

27 SS Управление конфликтующими типами памяти

28 HTT Поддержка Hyper-Threading

29 TM Поддержка автоматического мониторинга температуры

31 SBF Сигнал Останова при FERR

TLB and cache info:

b1: unknown TLB/cache descriptor

b0: дескриптор TLB-команд, 4K страницы, асс. 4-направ., 128 элементов

05: unknown TLB/cache descriptor

f0: unknown TLB/cache descriptor

57: unknown TLB/cache descriptor

56: unknown TLB/cache descriptor

78: unknown TLB/cache descriptor

30: L1 кэш-команд, 32 KB, асс. 8-направ., длина строки 64 байта

b4: unknown TLB/cache descriptor

2c: L1 кэш-данных, 32 KB, асс. 8-направ., длина строки 64 байта

Processor serial: 0001-067A-BFEB-FBFF-0400-E3BD