Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизації систем обробки інформації та управління

“ Оптимізація роботи з пам’яттю ”

Лабораторна робота №5

Перевірив: Дифучин А. Ю.

Виконав: Коноплянка Д. С.

Студент гр. ІС-73, ФІОТ,

3 курс

Київ

НТУУ «КПІ ім. І. Сікорського»

2020

**Тема:** оптимізація роботи з пам’яттю.

**Завдання на роботу:**

1. Проаналізувати програму, виміряти час її виконання, визначити, де є простір для поліпшення.
2. Виходячи з описаних у теорії механізмів організації пам'яті внести необхідні зміни в структуру коду.
3. Виміряти новий час виконання, за наявності очевидних поліпшень, переходити до п.4, якщо немає - до п.1
4. Зафіксувати нові показники і пояснити, за рахунок чого було досягнуто прискорення.

**Хід роботи:**

Під час виконання ЛР було написано програму на мові Java. В програмі створюється двомірний масив (матриця), по якому відбувається ітерація та акумуляція елементів в сумму.

Перший варіант - обхід за рядками, другий - за стовпцями.

Використовуючи програму на С++ ми займали кеш процесора іншими даними перед кожним запуском.

Використовуючи вбудовані засоби мови Java було здійснено замір часу при виконанні з закешованими даними та без.

**Лістинг програм:**

**Програма на С++, задача якої є зайняти кеш:**

#include <malloc.h>

#include <iostream>

using namespace std;

int main()

{

const int size = 2 \* 1024 \* 1024; // 2MB

char\* c = (char\*)malloc(size);

for (int i = 0; i < 0xffff; i++) {

for (int j = 0; j < size; j++) {

c[j] = i \* j;

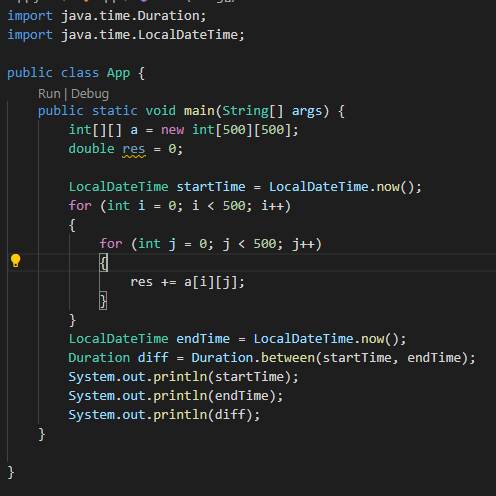
}

std::cout << i << "\n";

}

}

**Обхід матриці за рядками (Java):**

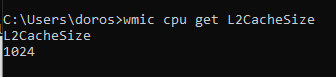


**Обхід матриці за стовпцями (Java):**



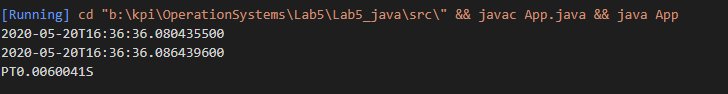
**Результати виконання:**

Розмірність L2CacheSize

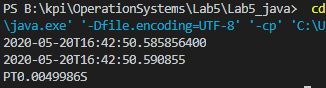


**Перший варіант програми:**

**З очищеним кешем:**

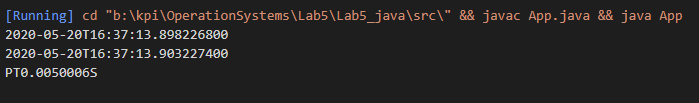


**Запуск з кешу:**

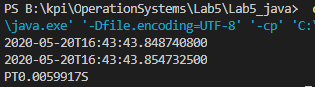


**Другий варіант програми:**

**З очищеним кешем:**



**Запуск з кешу:**



**Висновок:**

Виконання обох програм з кешу займає значно менший час, ніж тих самих програм після очищення кешу. Така оптимізація досягаєтся тим, що при повторному запуску програми вона використовує вже збережені в кеші дані, а не зчитує їх з пам’яті знову і знову.

Обхід за рядками займає менший час виконання, адже процес ітерації по двомірному масиву простіший, коли ми обходимо рядки, а не стовпці по черзі, все тому, що елементи масиву зберігаються в пам’яті один за одним

(мають чудову просторову локальність), навідміну від ітерації по стовпцям, де нам завжди треба стрибати по пам’яті, щоб доступитися до потрібної комірки.