Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизації систем обробки інформації та управління

“ Профілювання”

Лабораторна робота №6

Перевірив: Дифучин А. Ю.

Виконав: Коноплянка Д. С.

Студент гр. ІС-73, ФІОТ,

3 курс

Київ

НТУУ «КПІ ім. І. Сікорського»

2020

**Завдання на роботу:**

1) Проаналізувати програму, заміряти час її виконання, подивитися, де є простір для поліпшення.

2) Виходячи з описаних в теорії механізмів організації пам'яті внести необхідні зміни в структуру коду.

3) Заміряти новий час виконання, при 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**

Було проаналізовано код програми заданої по варіанту, використано утиліти gprof та time, потім було видалено зайві виклики sleep(1) та циклічні виклики функцій. Після цього до коду знову було застосовано gprof та time.

**Лістин програми**

#include<stdio.h>

#include "windows.h"

#include <iostream>

void new\_func1(void);

int resultOfSum(int a, int b)

{

return a + b;

}

int func1(int a, int b)

{

int res = 0;

int i;

for (i = 0; i < 10; i++)

{

Sleep(1);

if (i > 8)

res = resultOfSum(a, b);

if (res > 0)

return res;

}

return res;

}

int func2(int a, int b, int c)

{

int res = 0;

int i;

for (i = 0; i < 10; i++)

{

c++;

if (c > b)

res = func1(a, b);

else

res = func2(a, b, c);

if (res > 0)

return res;

}

return res;

}

int main(void)

{

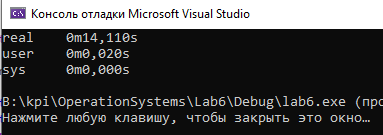
func2(35, 120, 28);

return 0;

}

**Результати утиліт gprof та time**





**Оптимізований код та результат утиліт** **gprof , time**

 #include<stdio.h>

int sum(int a, int b)

{

return a + b;

}

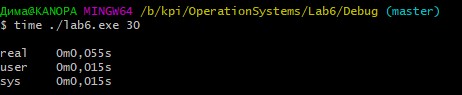
int main(void)

{

sum(51, 110);

return 0;

}



**Висновок**

Під час виконання роботи був вивчений інструмент профілювання програм gprof та проаналізований профіль двох версій однієї і тієї ж програми.

За допомогою утиліт **gprof , time** можна швидко та легко дізнатися час виконання програм та навіть певних функцій. Також **gprof** надає можливість побачити так званий «Граф викликів»