Київський національний університет імені Тараса Шевченка факультет радіофізики, електроніки та комп'ютерних систем

Лабораторна робота № 3

Тема: «Дослідження оптимізації коду з використанням векторних розширень CPU»

Роботу виконав
студент 2 курсу
КІ, СА
Здоров Ю.Д.

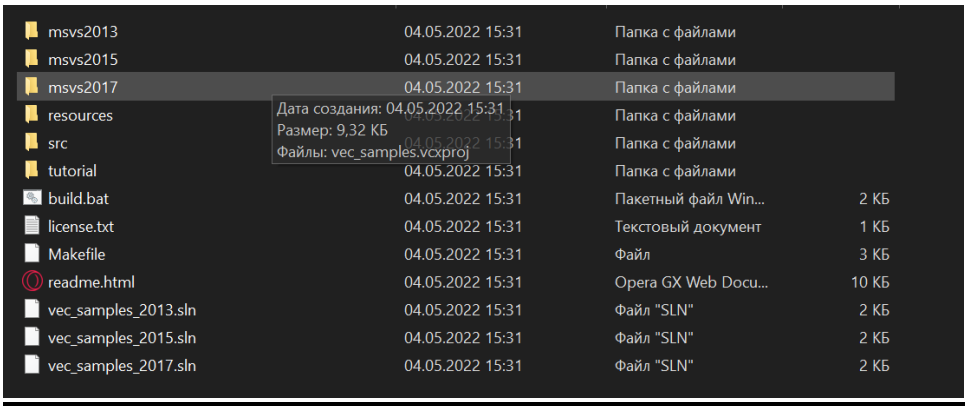
Київ 2021

Хід роботи

1. Отримайте доступ на обчислювальний кластер для роботи з Intel Compiler

DAC746B1B36BD16B	/C=UA/O=KNU/OU=People/L=FRECS/CN=Yuriy Zdorov	Tue, 27 Sep 2022 16:22:41

2. Завантажте файли Intel® C++ Compiler - Using Auto-Vectorization Tutorial



3. Використовуючи інструкції в readme.html ознайомились та виконали Tutorial на обчислювальному кластері

1. Встановлення змінних середовища:

Замість інструкцій в пункті “Setting the Environment Variables” завантажили оточення компілятора шляхом виконання команди: ml icc

```
KNU:  :s8 [tb358 ~]$ ml icc
KNU:  :s8 [tb358 ~]$ icc -01 -std=c99 Multiply.c Driver.c -o MatVector
```

2. Встановлення базової лінії продуктивності

```
KNU:  :s8 [tb358 ~]$ icc -01 -std=c99 Multiply.c Driver.c -o MatVector
KNU:  :s8 [tb358 ~]$ time ./MatVector

ROW:101 COL: 101
Execution time is 13.298 seconds
GigaFlops = 1.534240
Sum of result = 195853.999899

real    0m13.329s
user    0m13.291s
sys     0m0.006s
```

3. Створення звіту про векторізацію

```
KNU:  :s8 [tb358 ~]$ icc -std=c99 -O2 -D NOFUNCCALL -qopt-report=1 -qopt-repor
ply.c Driver.c -o MatVector
icc: remark #10397: optimization reports are generated in *.optrpt files in
on
KNU:  :s8 [tb358 ~]$ time ./MatVector

ROW:101 COL: 101
Execution time is 5.118 seconds
GigaFlops = 3.986127
Sum of result = 195853.999899

real    0m5.127s
```

```
KNU:  :s8 [tb358 ~]$ cat Driver.optrpt
Intel(R) Advisor can now assist with vectorizationand show optimization
  report messages with your source code.
See "http s://softwae.intelcom/en-us/intel-advisor-xe" for details.


Begin opti mization  report for: main()

    Report from: Vector optimizations [vec]


LOOP BEGIN at Driver.c(47,5) inlined into Driver.c(135,5)
   remark #25460: No loop optimizations reported

   LOOP BEGIN at Driver.c(48,9) inlined into Driver.c(135,5)
   <Peeled loop for vectorization>
   LOOP END

   LOOP BEGIN at Driver.c(48,9) inlined into Driver.c(135,5)
      remark #15300: LOOP WAS VECTORIZED
   LOOP END

   LOOP BEGIN at Driver.c(48,9) inlined into Driver.c(135,5)
   <Remainder loop for vectorization>
   LOOP END
LOOP END

LOOP BEGIN at Driver.c(62,5) inlined into Driver.c(136,5)
   remark #15300: LOOP WAS VECTORIZED
LOOP END
LOOP BEGIN at Driver.c(62,5) inlined into Driver.c(136,5)
   remark #15300: LOOP WAS VECTORIZED
LOOP END

LOOP BEGIN at Driver.c(62,5) inlined into Driver.c(136,5)
<Remainder loop for vectorization>
LOOP END

LOOP BEGIN at Driver.c(140,5)
   remark #25460: No loop optimizationsreported

   LOOP BEGIN at Driver.c(143,9)
      remark #25460: No loop optimizationsreported

      LOOP BEGIN at Driver.c(145,13)
         remark #15300: LOOP WAS VECTORIZED
      LOOP END

      LOOP BEGIN at Driver.c(145,13)
      <Remainder loop for vectorization>
      LOOP END
```

```
   LOOP END
LOOP END

LOOP BEGIN at Driver.c(74,5) inlined into Driver.c(159,5)
   remark #15300: LOOP WAS VECTORIZED
LOOP END

LOOP BEGIN at Driver.c(74,5) inlined into Driver.c(159,5)
<Remainder loop for vectorization>
LOOP END


Begin optimization report for: printsum(int, double *)

    Report from: Vecto r optimizations [vec]


LOOP BEGIN at Driver.c(74,5)
<Peeled loop for vectorization>
LOOP END

LOOP BEGIN at Driver.c(74,5)
   remark #15300: LOOP WAS VECTORIZED
LOOP END

LOOP BEGIN at Driver.c(74,5)
<Remainder loop for vectorization>
LOOP END


Begin optimization report for: init_array(int, double, doubl e *)

    Report from: Vector optimizations [vec]


LOOP BEGIN at Driver.c(62,5)
<Peeled loop for vectorization>
LOOP END

LOOP BEGIN at Driver.c(62,5)
   remark #15300: LOOP WAS VECTORIZED
LOOP END

LOOP BEGIN at Driver.c(62,5)
<Remainder loop for vectorization>
LOOP END
```

```
Begin optimization report for: init_matrix(int, int, double, double (*)[101])

    Report from: Vector optimizations [vec]


LOOP BEGIN at Driver.c(47,5)
   remark #25460: No loop optimizations reported

   LOOP BEGIN at Driver.c(48,9)
   <Peeled loop for vectorization>
   LOOP END

   LOOP BEGIN at Driver.c(48,9)
      remark #15300: LOOP WAS VECTORIZED
   LOOP END

   LOOP BEGIN at Driver.c(48,9)
   <Remainder loop for vectorization>
   LOOP END
LOOP END
```

```
KNU:  :s8 [tb358 ~]$ icc -std=c99 -O2 -D NOFUNCCALL -qopt-report-phase=vec,loop -qopt-report=2
Multiply.c Driver.c -o MatVector
icc: remark #10397: optimization reports are generated in *.optrpt files in the output locati
on
KNU:  :s8 [tb358 ~]$ cat Multiply.optrpt
Intel(R) Advisor can now assist with vectorization and show optimization
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for details.


Begin optimization report for: matvec(int, int, double (*)[*], double *, double *)

    Report from: Loop nest & Vector optimizations [loop, vec]


LOOP BEGIN at Multiply.c(37,5)
   remark #15541: outer loop was not auto-vectorized: consider using SIMD directive

   LOOP BEGIN at Multiply.c(49,9)
      remark #15344: loop was not vectorized: vector dependence prevents vectorization. First
 dependence is shown below. Use level 5 report for details
      remark #15346: vector dependence: assumed FLOW dependence between b[i] (50:13) and b[i]
 (50:13)
      remark #25439: unrolled with remainder by 2
   LOOP END

   LOOP BEGIN at Multiply.c(49,9)
   <Remainder>
   LOOP END
LOOP END
```

4. Підвищення продуктивності за допомогою розшифровки покажчика

```
KNU:  :s8 [tb358 ~]$ icc -std=c99 -qopt-report=2 -qopt-report-phase=vec -D NOALIAS Multiply.c D
river.c -o MatVector
icc: remark #10397: optimization reports are generated in *.optrpt files in the output locati
on
KNU:  :s8 [tb358 ~]$ time ./MatVector

ROW:101 COL: 101
Execution time is 5.519 seconds
GigaFlops = 3.696851
Sum of result = 195853.999899

real    0m5.525s
user    0m5.519s
sys     0m0.002s
KNU:  :s8 [tb358 ~]$ cat Multiply.optrpt
Intel(R) Advisor can now assist with vectorization and show optimization
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for details.


Begin optimization report for: matvec(int, int, double (*)[*], double *__restrict__, double *
)

    Report from: Vector optimizations [vec]


LOOP BEGIN at Multiply.c(37,5)
   remark #15542: loop was not vectorized: inner loop was already vectorized

   LOOP BEGIN at Multiply.c(49,9)
   <Peeled loop for vectorization>
   LOOP END

   LOOP BEGIN at Multiply.c(49,9)
      remark #15300: LOOP WAS VECTORIZED
   LOOP END

   LOOP BEGIN at Multiply.c(49,9)
   <Alternate Alignment Vectorized Loop>
   LOOP END

   LOOP BEGIN at Multiply.c(49,9)
   <Remainder loop for vectorization>
   LOOP END
LOOP END
```

5. Підвищення продуктивності шляхом вирівнювання даних
6.

```
KNU:  :s8 [tb358 ~]$ icc -std=c99 -qopt-report=4 -qopt-report-phase=vec -D NOALIAS -D ALIGNED M
ultiply.c Driver.c -o MatVector
icc: remark #10397: optimization reports are generated in *.optrpt files in the output locati
on
KNU:  :s8 [tb358 ~]$ time ./MatVector

ROW:101 COL: 102
Execution time is 5.093 seconds
GigaFlops = 4.005978
Sum of result = 195853.999899

real    0m5.219s
user    0m5.091s
sys     0m0.005s
```

```
KNU:  :s8 [tb358 ~]$ cat Multiply.optrpt
Intel(R) Advisor can now assist with vectorization and show optimization
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for details.

Intel(R) C Intel(R) 64 Compiler for applications running on Intel(R) 64, Version 18.0.5.274 B
uild 20180823

Compiler options: -std=c99 -qopt-report=4 -qopt-report-phase=vec -D NOALIAS -D ALIGNED -o Mat
Vector

Begin optimization report for: matvec(int, int, double (*)[*], double *__restrict__, double *
)

    Report from: Vector optimizations [vec]


LOOP BEGIN at Multiply.c(37,5)
   remark #15542: loop was not vectorized: inner loop was already vectorized

   LOOP BEGIN at Multiply.c(49,9)
      remark #15388: vectorization support: reference a[i][j] has aligned access   [ Multiply
.c(50,21) ]
      remark #15388: vectorization support: reference x[j] has aligned access   [ Multiply.c(
50,31) ]
      remark #15305: vectorization support: vector length 2
      remark #15399: vectorization support: unroll factor set to 4
      remark #15309: vectorization support: normalized vectorization overhead 0.594
      remark #15300: LOOP WAS VECTORIZED
      remark #15448: unmasked aligned unit stride loads: 2
      remark #15475: --- begin vector cost summary ---
      remark #15476: scalar cost: 10
      remark #15477: vector cost: 4.000
      remark #15478: estimated potential speedup: 2.410
      remark #15488: --- end vector cost summary ---
   LOOP END

   LOOP BEGIN at Multiply.c(49,9)
   <Remainder loop for vectorization>
      remark #15388: vectorization support: reference a[i][j] has aligned access   [ Multiply
.c(50,21) ]
      remark #15388: vectorization support: reference x[j] has aligned access   [ Multiply.c(
50,31) ]
      remark #15335: remainder loop was not vectorized: vectorization possible but seems inef
ficient. Use vector always directive or -vec-threshold0 to override
      remark #15305: vectorization support: vector length 2
```

7. Підвищення продуктивності за допомогою міжпроцедурної оптимізації

```
KNU:  :s8 [tb358 ~]$ icc -std=c99 -qopt-report=2 -qopt-report-phase=vec -D NOALIAS -D ALIGNED -
ipo Multiply.c Driver.c -o MatVector
icc: remark #10397: optimization reports are generated in *.optrpt files in the output locati
on
KNU:  :s8 [tb358 ~]$ time ./MatVector

ROW:101 COL: 102
Execution time is 4.892 seconds
GigaFlops = 4.170358
Sum of result = 195853.999899

real    0m4.898s
user    0m4.891s
sys     0m0.003s
```

```
KNU:  :s8 [tb358 ~]$ cat ipo_out.optrpt
Intel(R) Advisor can now assist with vectorization and show optimization
  report messages with your source code.
See "https://software.intel.com/en-us/intel-advisor-xe" for details.


Begin optimization report for: main()

    Report from: Vector optimizations [vec]


LOOP BEGIN at Driver.c(152,16)
   remark #15542: loop was not vectorized: inner loop was already vectorized

   LOOP BEGIN at Multiply.c(37,5) inlined into Driver.c(150,9)
      remark #15542: loop was not vectorized: inner loop was already vectorized

      LOOP BEGIN at Multiply.c(49,9) inlined into Driver.c(150,9)
         remark #15300: LOOP WAS VECTORIZED
      LOOP END

      LOOP BEGIN at Multiply.c(49,9) inlined into Driver.c(150,9)
      <Remainder loop for vectorization>
         remark #15335: remainder loop was not vectorized: vectorization possible but seems i
nefficient. Use vector always directive or -vec-threshold0 to override
      LOOP END
   LOOP END
LOOP END

LOOP BEGIN at Driver.c(74,5) inlined into Driver.c(159,5)
   remark #15300: LOOP WAS VECTORIZED
LOOP END

LOOP BEGIN at Driver.c(74,5) inlined into Driver.c(159,5)
<Remainder loop for vectorization>
LOOP END

Begin optimization report for: init_matrix(int, int, double, double (*)[102])

    Report from: Vector optimizations [vec]


LOOP BEGIN at Driver.c(47,5)
   remark #15542: loop was not vectorized: inner loop was already vectorized

   LOOP BEGIN at Driver.c(48,9)
      remark #15300: LOOP WAS VECTORIZED
   LOOP END

   LOOP BEGIN at Driver.c(48,9)
   <Remainder loop for vectorization>
   LOOP END
LOOP END

LOOP BEGIN at Driver.c(53,9)
   remark #15300: LOOP WAS VECTORIZED
LOOP END

LOOP BEGIN at Driver.c(53,9)
<Remainder loop for vectorization>
LOOP END
```

```
Begin optimization report for: init_array(int, double, double *)

    Report from: Vector optimizations [vec]


LOOP BEGIN at Driver.c(62,5)
   remark #15300: LOOP WAS VECTORIZED
LOOP END

LOOP BEGIN at Driver.c(62,5)
<Remainder loop for vectorization>
LOOP END
```

4. Оптимізація програми

1. Взяли неінтерактивну консольну програму мовою С, що обчислює факторіал при n=1000.

   Написали сценарій **scr.sh**, що:

a. Компілює програму з різними оптимізаціями (-O[1..3]) та виміряли час її роботи. Оскільки час досить малий - виміряли час роботи 1000 запусків алгоритму в циклі. Час роботи виміряли утилітою time.

```
Optimization O1
Execution time of 1000 launches:

real    0m2.807s
user    0m0.342s
sys     0m2.599s
Optimization O2
Execution time of 1000 launches:

real    0m2.864s
user    0m0.318s
sys     0m2.699s
Optimization O3
Execution time of 1000 launches:

real    0m2.900s
user    0m0.301s
sys     0m2.731s
```

b. Для кожного розширення компілює Intel-компілятором окремий варіант оптимізованого коду (-xExtension)
c. Отримує перелік всіх розширень процесору що підтримуються.
d. Вимірює час виконання кожного варіанта оптимізованої програми

Processor extensions:
Optimization SSE2
Execution time of 1000 launches:

real 0m2.953s
user 0m0.342s
sys 0m2.761s
Optimization SSE3
Execution time of 1000 launches:

real 0m3.017s
user 0m0.321s
sys 0m2.854s
Optimization SSSE3
Execution time of 1000 launches:

real 0m2.950s
user 0m0.344s
sys 0m2.761s
Optimization SSE4.1
Execution time of 1000 launches:

real 0m2.868s
user 0m0.333s
sys 0m2.691s
Optimization SSE4.2
Execution time of 1000 launches:

real 0m2.846s
user 0m0.322s
sys 0m2.668s
Optimization AVX
Execution time of 1000 launches:

real 0m2.879s
user 0m0.348s
sys 0m2.685s
Optimization CORE-AVX2

Please verify that both the operating system and the processor support Intel(R) MOVBE, F16C, FMA, BMI, LZCNT and AVX2 instructions.

Optimization CORE-AVX-I

Please verify that both the operating system and the processor support Intel(R) F16C instructions.

Optimization ATOM SSE4.2
Execution time o-f 1000launches:

real 0m2.832s
user 0m0.346s
sys 0m2.645s
Optimization ATOM SSSE3
Execution time ot-1000 launches:

real 0m2. 933s
user Om0.342s
sys 0m2.747s
Optimization MIC-AVX512

Please verify that both the operating system and the processor support Intel(R) MOVBE, FI6C, FMA, BMI, LZCNT, AVX2, AVX512F, ADX, AVX512ER, AVX512PF and AVX512CD instructions.

Optimization KNM

Please verify that both the operating system and the processor support Intel(R) MOVBE, FI6C, FMA, BMI, LZCNT, AVX2, AVX512F, ADX, AVX512ER, AVX512PF, AVX512CD, AVX512_4FMAPS, AVX512_4VNNI W and AVX512_VPOPCNTDQ instructions.

Optimization CORE-AVX512

Please verify that both the operating system and the processor support Intel(R) MOVBE, FI6C, FMA, BMI, LZCNT, AVX2, AVX512D0, AVX512F, ADX, AVX512CD, AVX512BW, AVX512VL and CLWB instructions.

Optimization COMMON-AVX512

Please verify that both the operating system and the processor support Intel(R) MOVBE, F16C, FMA, BMI, LZCNT, AVX2, AVX512F, AOX and AVX512CD instructions.

Optimization BROADWELL

Please verify that both the operating system and the processor support Intel(R) MOVBE, F16C, FMA, BMI, LZCNT, AVX2 and ADX instructions.

Optimization CANNONLAKE

Please verify that both the operating system and the processor support Intel(R) MOVBE, F16C, FMA, BMI, LZCNT, AVX2, AVX51200, AVX512F, ADX, AVX512IFMA52, AVX512CD, AVX512BW, AVX512VL and AVX512_VBMI instructions.

Optimization HASWELL

Please verify that both the operating system and the processor support Intel(R) MOVBE, F16C, FMA, BMI, LZCNT and AVX2 instructions.

Optimization ICELAKE-CLIENT

Please verify that both the operating system and the processor support Intel(R) MOVBE, F16C, FMA, BMI, LZCNT, AVX2, AVX512D0, AVX512F, AOX, AVXS12IFMA52, AVXS12CD, AVX5128W, AVX512VL, AVX512_VBMI, AVX512_VPOPCNTDQ, AVX512_BITALG, AVX512_VBMI2, GFNI, VAES, VPCLMULQDQ, AVX512_VNNI, CLWB and RDPID instructions.

Optimization ICELAKE-SERVER

Please verify that both the operating system and the processor support Intel(R) MOVBE, F16C, FMA, BMI, LZCNT, AVX2, AVX5120Q, AVX512F, AOX, AVX512IFMA52, AVX512CD, AVX5128W, AVX512VL, AVX512_VBMI, AVX512_VPOPCNTDQ, AVX512_BITALG, AVX512_VBMI2, GFNI, VAES, VPCLMULQDQ, AVX512_VNNI, CLWB and RDPID instructions.

Optimization IVYBRIDGE

```
Please verify that both the operating system and the processor support Intel(R) F16C instructions.

Optimization KNL

Please verify that both the operating system and the processor support Intel(R) MOVBE, F16C, FMA, BMI, LZCNT, AVX2, AVX512F, ADX, AVX512ER, AVX512PF and AVX512CD instructions.

Optimization SANDYBRIDGE
Execution time of 1000 launches:

real    0m2.887s
user    0m0.307s
sys     0m2.734s
Optimization SILVERMONT
Execution time of 1000 launches:

real    0m2.942s
user    0m0.355s
sys     0m2.757s
Optimization SKYLAKE

Please verify that both the operating system and the processor support Intel(R) MOVBE, F16C, FMA, BMI, LZCNT, AVX2 and ADX instructions.

Optimization SKYLAKE-AVX512

Please verify that both the operating system and the processor support Intel(R) MOVBE, F16C, FMA, BMI, LZCNT, AVX2, AVX512DQ, AVX512F, ADX, AVX512CD, AVX512BW, AVX512VL and CLWB instruction
s.
```

2. Запустили задачу в планувальник обчислювального кластеру 5 разів (для статистики на різних нодах)

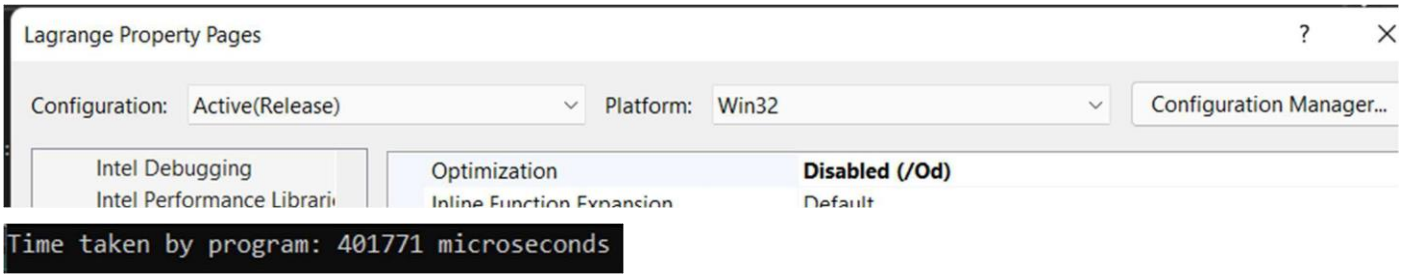
```
[tb358@plus7 ~]$ qsub -I -l nodes=1:ppn=1,walltime=00:30:00 src.sh
```

5. Встановили програмний продукт Intel® Parallel Studio

Оберали будь-який зі створених програмних продуктів та виконали його оптимізацію з використання Intel® Parallel Studio.

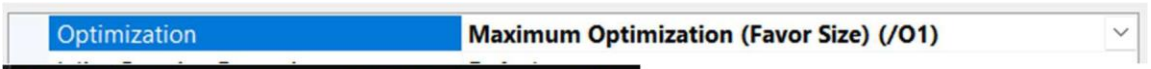
Використали програму Lagrange

1. Вимкнули будь-яку оптимізацію



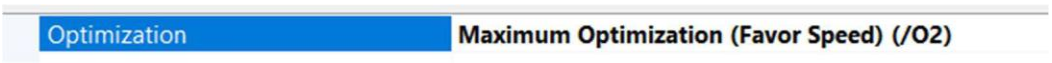

```
Time taken by program: 401771 microseconds
```

2. Оптимізація O1 – розмір вихідного коду



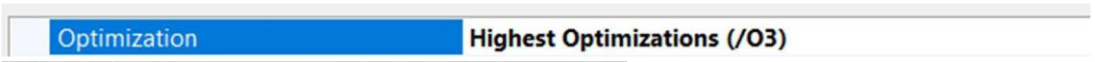

```
Time taken by program: 390706 microseconds
```

3. O2 – швидкість коду



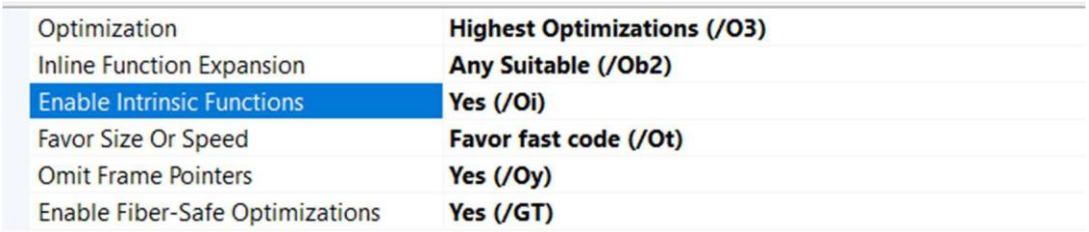

```
Time taken by program: 340260 microseconds
```

4. O3 – найбільша оптимізація




```
Time taken by program: 339734 microseconds
```

5. Увімкнули можливі оптимізації

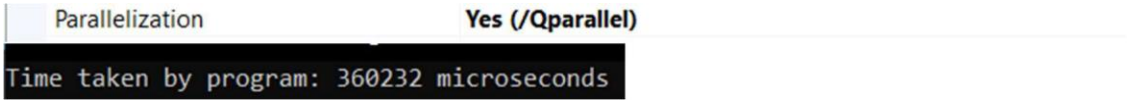



```
Time taken by program: 338409 microseconds
```

6. Увімкнули міжфайлову оптимізацію


```
Time taken by program: 346667 microseconds
```

7. Увімкнули розпаралелення




```
Time taken by program: 360232 microseconds
```

Висновки:

Репозиторій розміщення коду

https://github.com/Zapilman/KC_lab3

Виконуючи лабораторну роботу були опрацьовані навички роботи з intel C++

компілятором.

ICC компілятор дозволяє компілювати C/C++ програми з використанням різних типів оптимізації та певних розширень, що дозволяють краще працювати з певними процесорами. Оптимізація O1 передбачає максимал скорочення коду та виконуваних інструкцій, що дозволяє зменшувати використовуване місце в оперативній пам'яті та кеші процесора. Оптиціsація O2 передбачає таку компіляцію коду, яка дозволить процесору якнайшвидше виконати вказані інструкції, що взагалі впливає на продуктивність роботи системи та витрачених ресурсів. Оптимізація O3 поєднує в собі обидва попередні типи, що охоплює усі аспекти оптимізації ПЗ архітектури Intel. Основою такої оптимізації є векторизація, коли інформація обробляється не поодинокими шматками, а певним. Також при роботі з Intel Parallel Studio зробили певні висноки, а саме: Intel Parallel Studio є інтегрованим ПЗ в середовище Visual Studio, що дозволяє більш комфортно компілювати проєкти за допомогою Intel C/C++ Compiler, та зберігає функціонал ICC компілятора Unix-системи.