МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Качество и метрология программного обеспечения»

ТЕМА: «Измерение характеристик динамической сложности программ с помощью профилировщика SAMPLER»

Студентка гр. 6304 Иванкова В.М.

Преподаватель Кирьянчиков В.А.

Санкт-Петербург

2020

Задание

1. Ознакомиться с документацией на монитор SAMPLER и выполнить под его управлением тестовые программы test_cyc.cpp и test_sub.cpp с анализом параметров повторения циклов, структуры описания циклов, способов профилирования процедур и проверкой их влияния на точность и чувствительность профилирования.

2. Скомпилировать и выполнить под управлением SAMPLER'a программу на C, разработанную в 1-ой лабораторной работе.
Выполнить разбиение программы на функциональные участки и снять профили для двух режимов:

1 - измерение только полного времени выполнения программы;

2 - измерение времен выполнения функциональных участков (ФУ).

Убедиться, что сумма времен выполнения ФУ соответствует полному времени выполнения программы.

3. Выявить "узкие места", связанные с ухудшением производительности программы, ввести в программу усовершенствования и получить новые профили. Объяснить смысл введенных модификаций программ.

Ход работы

Использовался старый SAMPLER. Программы компилировались с помощью Borland C++. Компилирование выполнялось на Windows XP, профилирование – в DOSBox.

Тестовые программы

Код программы test_cyc.cpp с нумерацией строк представлен в приложении А.

Результаты профилирования:

```
---------------------------------------------------------------------
   NN            Имя обработанного файла
---------------------------------------------------------------------
   1.       ..\TEST\TEST_CYC.CPP
---------------------------------------------------------------------

   Таблица с результатами измерений ( используется 13 из 416 записей )

---------------------------------------------------------------------
```

```
 Исх.Поз. Прием.Поз.  Общее время(мкс)  Кол-во прох. Среднее время(мкс)
 ------------------------------------------------------------------------
  1 :   8  1 :  10            4335.47            1            4335.47
 ------------------------------------------------------------------------
  1 :  10  1 :  12            8675.14            1            8675.14
 ------------------------------------------------------------------------
  1 :  12  1 :  14           21671.50            1           21671.50
 ------------------------------------------------------------------------
  1 :  14  1 :  16           43348.03            1           43348.03
 ------------------------------------------------------------------------
  1 :  16  1 :  19            4335.47            1            4335.47
 ------------------------------------------------------------------------
  1 :  19  1 :  22            8670.11            1            8670.11
 ------------------------------------------------------------------------
  1 :  22  1 :  25           21678.20            1           21678.20
 ------------------------------------------------------------------------
  1 :  25  1 :  28           43343.00            1           43343.00
 ------------------------------------------------------------------------
  1 :  28  1 :  34            4335.47            1            4335.47
 ------------------------------------------------------------------------
  1 :  34  1 :  40            8675.98            1            8675.98
 ------------------------------------------------------------------------
  1 :  40  1 :  46           21671.50            1           21671.50
 ------------------------------------------------------------------------
  1 :  46  1 :  52           43348.03            1           43348.03
 ------------------------------------------------------------------------
-------------------------------------------------------------------------
```

По результатам видно, что времена сильно завышены из-за накладных затрат эмулятора. В коде используется разная запись циклов с одинаковым количеством итераций, при этом отсутствует влияние на время. А также видна линейная зависимость времени от количества итераций.

Код программы test_sub.cpp с нумерацией строк представлен в приложении Б.

Результаты профилирования:

```
-------------------------------------------------------------------------
  NN            Имя обработанного файла
-------------------------------------------------------------------------
  1.  ..\TEST\TEST_SUB.CPP
-------------------------------------------------------------------------


  Таблица с результатами измерений ( используется 5 из 416 записей )

-------------------------------------------------------------------------
Исх.Поз. Прием.Поз.  Общее время(мкс)  Кол-во прох. Среднее время(мкс)
-------------------------------------------------------------------------
 1 :  30  1 :  32          433697.35            1          433697.35
-------------------------------------------------------------------------
 1 :  32  1 :  34          867392.18            1          867392.18
-------------------------------------------------------------------------
 1 :  34  1 :  36         2168468.29            1         2168468.29
-------------------------------------------------------------------------
 1 :  36  1 :  38         4336936.59            1         4336936.59
-------------------------------------------------------------------------
```

По результатам можно сделать аналогичные выводы о том, что время выполнения:

1) линейно зависит от количества итераций цикла;

2) сильно завышено из-за накладных затрат эмулятора.

Программа из первой лабораторной работы

Код программы из первой лабораторной работы с нумерацией строк представлен в приложениях В (для измерения полного времени) и Г (для измерения времен выполнения ФУ).

Результаты профилирования с измерением полного времени:

```
-----------------------------------------------------------------------
   NN            Имя обработанного файла
-----------------------------------------------------------------------
   1.      ..\TEST\LAB3_1.CPP
-----------------------------------------------------------------------


   Таблица с результатами измерений ( используется 3 из 416 записей )

-----------------------------------------------------------------------
Исх.Поз. Прием.Поз.  Общее время(мкс)  Кол-во прох. Среднее время(мкс)
-----------------------------------------------------------------------
 1 :  53  1 :  55           10026.15             1           10026.15
-----------------------------------------------------------------------
 1 :  55  1 :  57           29683.70             1           29683.70
```

Общее время выполнения первой функции – 10026,15 мкс, второй – практически в три раза больше и составляет 29683,70 мкс. Результаты также завышены из-за накладных затрат эмулятора.

Результаты профилирования с измерением времен ФУ:

```
-----------------------------------------------------------------------
   NN            Имя обработанного файла
-----------------------------------------------------------------------
   1.  ..\TEST\LAB3_2.CPP
-----------------------------------------------------------------------


   Таблица с результатами измерений ( используется 13 из 416 записей )

-----------------------------------------------------------------------
Исх.Поз. Прием.Поз.  Общее время(мкс)  Кол-во прох. Среднее время(мкс)
-----------------------------------------------------------------------
 1 :  17  1 :  20             2.51             1              2.51
-----------------------------------------------------------------------
 1 :  20  1 :  23            10.90             6              1.82
-----------------------------------------------------------------------
 1 :  23  1 :  26           916.88           402              2.28
-----------------------------------------------------------------------
 1 :  26  1 :  29          2631.62           356              7.39
 1 :  26  1 :  34          1818.67           251              7.25
-----------------------------------------------------------------------
 1 :  29  1 :  40            94.70           356              0.27
-----------------------------------------------------------------------
 1 :  34  1 :  36          3572.81           251             14.23
```

```
-----------------------------------------------------------------------
 1 :  36  1 :  38               0.00           251               0.00
-----------------------------------------------------------------------
 1 :  38  1 :  40              72.08            46               1.57
 1 :  38  1 :  26             397.26           205               1.94
-----------------------------------------------------------------------
 1 :  40  1 :  23             621.03           396               1.57
 1 :  40  1 :  42               5.87             6               0.98
-----------------------------------------------------------------------
 1 :  42  1 :  20              13.41             5               2.68
 1 :  42  1 :  44               2.51             1               2.51
-----------------------------------------------------------------------
 1 :  44  1 :  50               4.19             1               4.19
-----------------------------------------------------------------------
 1 :  50  1 :  55           29723.93             1           29723.93
-----------------------------------------------------------------------
```

По результатам измерений времени на ФУ видно, что время выполнения первой функции – 10152,44 мкс, второй – 29723,93 мкс. Данные времена практически не отличаются от полученных ранее, тем не менее время выполнения второй функции по-прежнему в 3 раза больше первой. Одной из причин является вызов функции *swap* внутри первой функции. Для усовершенствования производительности заменим вызов функции на её содержимое.

Измененная программа из первой лабораторной работы

Измененный код программы из первой лабораторной работы с нумерацией строк представлен в приложениях Д (для измерения полного времени) и Е (для измерения времен выполнения ФУ).

Результаты профилирования с измерением полного времени:

```
-----------------------------------------------------------------------
   NN            Имя обработанного файла
-----------------------------------------------------------------------
   1.      ..\TEST\LAB3_3.CPP
-----------------------------------------------------------------------


   Таблица с результатами измерений ( используется 3 из 416 записей )

-----------------------------------------------------------------------
Исх.Поз. Прием.Поз.  Общее время(мкс)  Кол-во прох. Среднее время(мкс)
-----------------------------------------------------------------------
 1 :  50  1 :  52           10046.26             1           10046.26
-----------------------------------------------------------------------
 1 :  52  1 :  54           29713.87             1           29713.87
```

Общее время выполнения первой и второй функций не изменилось и составляет 10046,26 мкс и 29713,87 мкс соответственно. Результаты также завышены из-за накладных затрат эмулятора.

Результаты профилирования с измерением времен ФУ:

NN	Имя обработанного файла
1.	..\TEST\LAB3_4_1.CPP

Таблица с результатами измерений (используется 13 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 9	1 : 12	2.51	1	2.51
1 : 12	1 : 15	11.73	6	1.96
1 : 15	1 : 18	902.63	402	2.25
1 : 18	1 : 21	2650.06	356	7.44
1 : 18	1 : 26	1820.35	251	7.25
1 : 21	1 : 34	96.38	356	0.27
1 : 26	1 : 30	3269.41	251	13.03
1 : 30	1 : 32	0.00	251	0.00
1 : 32	1 : 34	71.24	46	1.55
1 : 32	1 : 18	388.04	205	1.89
1 : 34	1 : 15	639.47	396	1.61
1 : 34	1 : 36	7.54	6	1.26
1 : 36	1 : 12	14.25	5	2.85
1 : 36	1 : 38	2.51	1	2.51
1 : 38	1 : 44	5.03	1	5.03
1 : 44	1 : 49	29619.17	1	29619.17

По результатам измерений времени на ФУ видно, что время выполнения первой функции – 9881,15 мкс, второй – 29619,17 мкс. В результате можно сделать выводы, что время первой функции уменьшилось после изменения примерно на 10%, а результаты также завышены из-за накладных затрат эмулятора.

Выводы

В результате выполнения данной лабораторной работы был изучен монитор SAMPLER, с помощью которого было выполнено профилирование тестовых программ test_cyc.cpp и test_sub.cpp.

Было проанализировано полное время выполнения программы, разработанной в 1-ой лабораторной работе, и время выполнения её ФУ.

Удалось частично усовершенствовать производительность программы из 1-ой лабораторной работы за счёт удаления внутреннего вызова функции *swap*.

ПРИЛОЖЕНИЕ А

TEST_CYC.C

```
1 #include <stdlib.h>
2 #include "Sampler.h"
3 #define Size 10000
4 int i, tmp, dim[Size];
5
6 void main()
7 {
8       SAMPLE;
9       for(i=0;i<Size/10;i++){ tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
10      SAMPLE;
11      for(i=0;i<Size/5;i++){ tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
12      SAMPLE;
13      for(i=0;i<Size/2;i++){ tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
14      SAMPLE;
15      for(i=0;i<Size;i++)  { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
16      SAMPLE;
17      for(i=0;i<Size/10;i++)
18        { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
19      SAMPLE;
20     for(i=0;i<Size/5;i++)
21       { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
22     SAMPLE;
23    for(i=0;i<Size/2;i++)
24       { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
25     SAMPLE;
26     for(i=0;i<Size;i++)
27       { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
28     SAMPLE;
29     for(i=0;i<Size/10;i++)
30       { tmp=dim[0];
31         dim[0]=dim[i];
32         dim[i]=tmp;
33       };
34     SAMPLE;
35     for(i=0;i<Size/5;i++)
36      { tmp=dim[0];
37         dim[0]=dim[i];
38         dim[i]=tmp;
39       };
40     SAMPLE;
41     for(i=0;i<Size/2;i++)
42       { tmp=dim[0];
43         dim[0]=dim[i];
44         dim[i]=tmp;
45       };
46     SAMPLE;
47     for(i=0;i<Size;i++)
48       { tmp=dim[0];
49         dim[0]=dim[i];
50         dim[i]=tmp;
51       };
52     SAMPLE;
53 }
```

ПРИЛОЖЕНИЕ Б

TEST_SUB.C

```
1 #include <stdlib.h>
2 #include "Sampler.h"
3 const unsigned Size = 1000;
4
5
6 void TestLoop(int nTimes)
7 {
8   static int TestDim[Size];
9   int tmp;
10   int iLoop;
11
12   while (nTimes > 0)
13   {
14     nTimes --;
15
16    iLoop = Size;
17   while (iLoop > 0)
18   {
19      iLoop -- ;
20      tmp = TestDim[0];
21      TestDim[0] = TestDim[nTimes];
22    TestDim[nTimes] = tmp;
23    }
24   }
25 } /* TestLoop */
26
27
28 void main()
29 {
30   SAMPLE;
31   TestLoop(Size / 10); // 100 * 1000  ¯®¢â®à¥•¨©
32   SAMPLE;
33   TestLoop(Size / 5);  // 200 * 1000  ¯®¢â®à¥•¨©
34   SAMPLE;
35   TestLoop(Size / 2);  // 500 * 1000  ¯®¢â®à¥•¨©
36   SAMPLE;
37   TestLoop(Size / 1);  // 1000* 1000  ¯®¢â®à¥•¨©
38   SAMPLE;
39 }
```

ПРИЛОЖЕНИЕ В

Полное время LAB3_1.C

```
1  #include <stdio.h>
2  #include <time.h>
3  #include <stdlib.h>
4  #include "Sampler.h"
5
6  void swap (float *x, float *y)
7  {
8         float temp;
9         temp = *x;
10        *x = *y;
11        *y = temp;
12 }
13
14 void shellsort(float arr[], int num)
15 {
16     int i, j, k;
17     for (i = num / 2; i > 0; i = i / 2)
18     {
19         for (j = i; j < num; j++)
20         {
21             for(k = j - i; k >= 0; k = k - i)
22             {
23                 if (arr[k+i] >= arr[k])
24                     break;
25                 else
26                 {
27                     swap(&arr[k], &arr[k+i]);
28                 }
29             }
30         }
31     }
32 }
33
34 void write_arr(float arr[], int num)
35 {
36        int i;
37        for (i = 0; i < num; i++)
38        {
39          printf("%f ", arr[i]);
40     }
41 }
42 int main()
43 {
44        int num = 80;
45        float my_max = 100.0;
46     float arr[80];
47     int k;
48
49     for (k = 0 ; k < num; k++)
50     {
51         arr[k] = (float)rand()/(float)(RAND_MAX/my_max);
52     }
53        SAMPLE;
54     shellsort(arr, num);
55        SAMPLE;
56     write_arr(arr, num);
57        SAMPLE;
58     return 0;
59 }
```

ПРИЛОЖЕНИЕ Г

Время ФУ LAB3_2.C

```
1  #include <stdio.h>
2  #include <time.h>
3  #include <stdlib.h>
4  #include "Sampler.h"
5
6  void swap (float *x, float *y)
7  {
8        float temp;
9        temp = *x;
10       *x = *y;
11       *y = temp;
12 }
13
14 void shellsort(float arr[], int num)
15 {
16     int i, j, k;
17       SAMPLE;
18     for (i = num / 2; i > 0; i = i / 2)
19     {
20       SAMPLE;
21         for (j = i; j < num; j++)
22         {
23              SAMPLE;
24             for(k = j - i; k >= 0; k = k - i)
25             {
26                     SAMPLE;
27                 if (arr[k+i] >= arr[k])
28                           {
29                           SAMPLE;
30                     break;
31                           }
32                 else
33                 {
34                           SAMPLE;
35                     swap(&arr[k], &arr[k+i]);
36                           SAMPLE;
37                 }
38                           SAMPLE;
39             }
40                     SAMPLE;
41         }
42              SAMPLE;
43     }
44       SAMPLE;
45 }
46
47 void write_arr(float arr[], int num)
48 {
49       int i;
50       SAMPLE;
51       for (i = 0; i < num; i++)
52       {
53         printf("%f ", arr[i]);
54     }
55       SAMPLE;
56 }
57 int main()
58 {
59       int num = 80;
60       float my_max = 100.0;
61     float arr[80];
```

```
62      int k;
63
64      for (k = 0 ; k < num; k++)
65      {
66          arr[k] = (float)rand()/(float)(RAND_MAX/my_max);
67      }
68      shellsort(arr, num);
69      write_arr(arr, num);
70      return 0;
71  }
```

ПРИЛОЖЕНИЕ Д

Полное время измененной LAB3_3.C

```
1  #include <stdio.h>
2  #include <time.h>
3  #include <stdlib.h>
4  #include "Sampler.h"
5
6  void shellsort(float arr[], int num)
7  {
8      int i, j, k;
9      for (i = num / 2; i > 0; i = i / 2)
10     {
11         for (j = i; j < num; j++)
12         {
13             for(k = j - i; k >= 0; k = k - i)
14             {
15                 if (arr[k+i] >= arr[k])
16
17                     break;
18
19                 else
20                 {
21                     float temp = arr[k+i];
22                     arr[k+i] = arr[k];
23                     arr[k] = temp;
24                 }
25             }
26         }
27     }
28
29 }
30
31 void write_arr(float arr[], int num)
32 {
33
34
35
36         printf("%f ", arr[i]);
37     }
38 }
39 int main()
40 {
41
42
43     float arr[80];
44     int k;
45
46     for (k = 0 ; k < num; k++)
47     {
48         arr[k] = (float)rand()/(float)(RAND_MAX/my_max);
49     }
50
51     shellsort(arr, num);
52
53     write_arr(arr, num);
54
55     return 0;
56 }
```

ПРИЛОЖЕНИЕ Е

Время ФУ измененной LAB3_4_1.C

```
1  #include <stdio.h>
2  #include <time.h>
3  #include <stdlib.h>
4  #include "Sampler.h"
5
6  void shellsort(float arr[], int num)
7  {
8      int i, j, k;
9        SAMPLE;
10     for (i = num / 2; i > 0; i = i / 2)
11     {
12       SAMPLE;
13         for (j = i; j < num; j++)
14         {
15              SAMPLE;
16             for(k = j - i; k >= 0; k = k - i)
17             {
18                     SAMPLE;
19                 if (arr[k+i] >= arr[k])
20                         {
21                         SAMPLE;
22                     break;
23                         }
24                 else
25                 {
26                         SAMPLE;
27                     float temp = arr[k+i];
28                     arr[k+i] = arr[k];
29                     arr[k] = temp;
30                                 SAMPLE;
31                 }
32                         SAMPLE;
33             }
34                     SAMPLE;
35         }
36              SAMPLE;
37     }
38       SAMPLE;
39 }
40
41 void write_arr(float arr[], int num)
42 {
43       int i;
44       SAMPLE;
45       for (i = 0; i < num; i++)
46       {
47         printf("%f ", arr[i]);
48     }
49       SAMPLE;
50 }
51 int main()
52 {
53       int num = 80;
54       float my_max = 100.0;
55     float arr[80];
56     int k;
57
58     for (k = 0 ; k < num; k++)
59     {
60         arr[k] = (float)rand()/(float)(RAND_MAX/my_max);
61     }
62     shellsort(arr, num);
```

```
63      write_arr(arr, num);
64      return 0;
65  }
```