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»

Студент гр. 6304 _____ Некрасов Н.А.

Преподаватель _____ Кирьянчиков В.А.

Санкт-Петербург

2020

Задание

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

1. Скомпилировать и выполнить под управлением SAMPLER'a программу на С, 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 Ubuntu с использованием Wine и DosBox.

Тестовые программы

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

Результаты профилирования:

```
-------------------------------------------------------------------
  NN          И м я  о б р а б о т а н н о г о  ф а й л а
-------------------------------------------------------------------
  1. 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            8675.98            1           8675.98
-----------------------------------------------------------------------
 1 :  12  1 :  14           21671.50            1          21671.50
-----------------------------------------------------------------------
 1 :  14  1 :  16           43348.87            1          43348.87
-----------------------------------------------------------------------
 1 :  16  1 :  19            4337.15            1           4337.15
-----------------------------------------------------------------------
 1 :  19  1 :  22            8668.43            1           8668.43
-----------------------------------------------------------------------
 1 :  22  1 :  25           21672.34            1          21672.34
-----------------------------------------------------------------------
 1 :  25  1 :  28           43348.03            1          43348.03
-----------------------------------------------------------------------
 1 :  28  1 :  34            4334.64            1           4334.64
-----------------------------------------------------------------------
 1 :  34  1 :  40            8670.11            1           8670.11
-----------------------------------------------------------------------
 1 :  40  1 :  46           21676.53            1          21676.53
-----------------------------------------------------------------------
 1 :  46  1 :  52           43348.87            1          43348.87
```

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.c с нумерацией строк представлен в приложении Б.

Результаты профилирования:

```
-----------------------------------------------------------------------
  NN          Имя обработанного файла
-----------------------------------------------------------------------
  1. 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          433699.86            1         433699.86
-----------------------------------------------------------------------
 1 :  32  1 :  34          867392.18            1         867392.18
-----------------------------------------------------------------------
```

```
 1 :  34  1 :  36         2168480.87            1        2168480.87
-------------------------------------------------------------------
 1 :  36  1 :  38         4336949.16            1        4336949.16
-------------------------------------------------------------------
```

По результатам можно сделать аналогичные выводы о том, что время выполнения линейно зависит от количества итераций цикла и завышено ввиду использования эмулятора

Программа из ЛР1.

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. LR1.CPP
-------------------------------------------------------------------

  Таблица с результатами измерений ( используется 3
  из 416 записей )

-------------------------------------------------------------------
   Исх.Поз. Прием.Поз.       Общее время(мкс)
                Кол-во прох. Среднее время(мкс)
-------------------------------------------------------------------
 1 :  31  1 :  35         5198277.71            1        5198277.71
```

Общее время выполнения первой функции – 5198278 мкс. Результаты завышены из-за затрат на работу эмулятора, а также тем, что ноутбук был дважды залит водой.

Результаты профилирования с измерением времен ФУ:

```
-------------------------------------------------------------------
  NN          Имя обработанного файла
-------------------------------------------------------------------
  1. LR1.CPP
-------------------------------------------------------------------

          Таблица с результатами измерений (
                используется 17 из 416 записей )
-------------------------------------------------------------------
   Исх.Поз. Прием.Поз.       Общее время(мкс)
                Кол-во прох. Среднее время(мкс)
-------------------------------------------------------------------
 1 :   9  1 :  11               3.35            1              3.35
```

```
-----------------------------------------------------------------------
 1 :  11  1 :  13            3438.71           4858              3.44
-----------------------------------------------------------------------
 1 :  13  1 :  15         6797649.14           4858            147.52
-----------------------------------------------------------------------
 1 :  15  1 :  19         7090481.74         11802511          153.87
-----------------------------------------------------------------------
 1 :  19  1 :  21          543642.36          654242            42.97

-----------------------------------------------------------------------
 1 :  21  1 :  23         7446104.36         4858              214.97
-----------------------------------------------------------------------
 1 :  39  1 :  43        50578720.02         1            50578720.02
-----------------------------------------------------------------------
1 :   35  1 :  39        5.35                1                  5.35
-----------------------------------------------------------------------
```

По результатам измерений времени на ФУ видно, что время выполнения первой функции – 50578720.73 мкс.

Измененная программа из первой лабораторной работы

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

Результаты профилирования с измерением полного времени:

```
-----------------------------------------------------------------------
  NN          Имя обработанного файла
-----------------------------------------------------------------------
  1. LR1.CPP
-----------------------------------------------------------------------
   Таблица с результатами измерений (
   используется 3 из 416 записей )

  -----------------------------------------------------------------------
Исх.Поз. Прием.Поз.       Общее время(мкс)
                Кол-во прох. Среднее время(мкс)
  -----------------------------------------------------------------------
   1 :  37  1 :  41         4662712.50           1        5180792.50
  -----------------------------------------------------------------------
```

Общее время выполнения первой функции уменьшилось примерно на 10% и стало 4662712.8. Вероятная причина такого поведения - вставка кода функции swap в тело функции, из-за чего не потребовалось дополнительное время на обращение к данной функции.

Результаты профилирования с измерением времен ФУ:

```
---------------------------------------------------------------------
   NN          Имя обработанного файла
---------------------------------------------------------------------
   2. LR1.CPP
---------------------------------------------------------------------

          Таблица с результатами измерений (
                используется 17 из 416 записей )
---------------------------------------------------------------------
    Исх.Поз. Прием.Поз.       Общее время(мкс)
                Кол-во прох. Среднее время(мкс)
---------------------------------------------------------------------
 1 :   9  1 :  11              3.35           1              3.35
---------------------------------------------------------------------
 1 :  11  1 :  13           3438.71          4858              3.44
---------------------------------------------------------------------
 1 :  13  1 :  15         6797649.14          4858            147.52
---------------------------------------------------------------------
 1 :  15  1 :  17         7090481.74         11802511         153.87
---------------------------------------------------------------------
 1 :  17  1 :  19         5436104.36         43251             42.94
---------------------------------------------------------------------
 1 :  19  1 :  21         542351.36          43251             42.12
---------------------------------------------------------------------
 1 :  21  1 :  23         4532104.12         43251             74.87
---------------------------------------------------------------------
 1 :  23  1 :  25         2744104.02         4858              80.23
---------------------------------------------------------------------
 1 :  25  1 :  27         3453612.36         4858             246.56
---------------------------------------------------------------------
 1 :  27  1 :  29         183583.36          4858             127.36
---------------------------------------------------------------------
1 :   39  1 :  43        4662809.02          1            4662809.02
---------------------------------------------------------------------
1 :   35  1 :  39        5.35               1                5.35

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

По результатам измерений времени на ФУ видно, что время выполнения функции составил 4662809.04 мкс.

Выводы

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

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

ПРИЛОЖЕНИЕ А

TEST_CYC.C

```
1. #define Size 10000
2. int i, tmp, dim[Size];
3.
4. void main()
5. {
6.      for(i=0;i<Size/10;i++){ tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
7.      for(i=0;i<Size/5;i++){ tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
8.      for(i=0;i<Size/2;i++){ tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
9.      for(i=0;i<Size;i++)  { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
10.        for(i=0;i<Size/10;i++)
11.          { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
12.        for(i=0;i<Size/5;i++)
13.          { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
14.        for(i=0;i<Size/2;i++)
15.          { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
16.        for(i=0;i<Size;i++)
17.          { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
18.        for(i=0;i<Size/10;i++)
19.          { tmp=dim[0];
20.            dim[0]=dim[i];
21.            dim[i]=tmp;
22.          };
23.        for(i=0;i<Size/5;i++)
24.          { tmp=dim[0];
25.            dim[0]=dim[i];
26.            dim[i]=tmp;
27.          };
28.        for(i=0;i<Size/2;i++)
29.          { tmp=dim[0];
30.            dim[0]=dim[i];
31.            dim[i]=tmp;
32.          };
33.        for(i=0;i<Size;i++)
34.          { tmp=dim[0];
35.            dim[0]=dim[i];
36.            dim[i]=tmp;
37.          };
38.   }
```

ПРИЛОЖЕНИЕ Б

TEST_SUB.C

```
1. const unsigned Size = 1000;
2.
3.
4. void TestLoop(int nTimes)
5. {
6.   static int TestDim[Size];
7.   int tmp;
8.   int iLoop;
9.
10.  while (nTimes > 0)
11.  {
12.    nTimes --;
13.
14.    iLoop = Size;
15.    while (iLoop > 0)
16.    {
17.      iLoop -- ;
18.      tmp = TestDim[0];
19.      TestDim[0] = TestDim[nTimes];
20.      TestDim[nTimes] = tmp;
21.    }
22.  }
23.} /* TestLoop */
24.
25.
26.void main()
27.{
28.   TestLoop(Size / 10); // 100 * 1000  п о в т о р е н и й
29.   TestLoop(Size / 5);  // 200 * 1000  п о в т о р е н и й
30.   TestLoop(Size / 2);  // 500 * 1000  п о в т о р е н и й
31.   TestLoop(Size / 1);  // 1000* 1000  п о в т о р е н и й
32.}
```

ПРИЛОЖЕНИЕ В

Полное время LR.C

```
1. #include <stdio.h>
2. #include <time.h>
3. #include <stdlib.h>
4. #include "sampler.h"
5.
6. #define MAX 4859
7.
8. void selectionSort(int array[], int size) {
9. SAMPLE;
10. for (int i = 0; i < size - 1; i++) {
11. SAMPLE;
12. int maxIndex = i;
13. SAMPLE;
14. for (int j = i + 1; j < size; j++) {
15. SAMPLE;
16. if (array[j] > array[maxIndex]){
17. maxIndex = j;
18. }
19. SAMPLE;
20. }
21. SAMPLE;
22. swap(&array[i], &array[maxIndex]);
23. SAMPLE;
24. }
25. }
26.
27. void swap(int* a, int* b){
28. int temp = *a;
29. *a = *b;
30. *b = temp;
31. }
32.
33. int main()
34. {
35. SAMPLE;
36. srand(time(NULL));
37. int array[MAX];
38.
39. SAMPLE;
40. for(int i = 0; i < MAX; ++i)
41. array[i] = rand();
42. selectionSort(array, MAX);
43. SAMPLE;
44.
45. return 0;
46. }
```

ПРИЛОЖЕНИЕ Г

LR.C отдельные замеры

```
1. #include <stdio.h>
2. #include <time.h>
3. #include <stdlib.h>
4. #include "sampler.h"
5.
6. #define MAX 4859
7.
8. void selectionSort(int array[], int size) {
9. SAMPLE;
10. for (int i = 0; i < size - 1; i++) {
11. SAMPLE;
12. int maxIndex = i;
13. SAMPLE;
14. for (int j = i + 1; j < size; j++) {
15. SAMPLE;
16. if (array[j] > array[maxIndex]){
17. maxIndex = j;
18. }
19. SAMPLE;
20. }
21. SAMPLE;
22. swap(array[i], array[maxIndex]);
23. SAMPLE;
24. }
25. }
26.
27. void swap(int* a, int* b){
28. int temp = *a;
29. *a = *b;
30. *b = temp;
31. }
32.
33. int main()
34. {
35. SAMPLE;
36. srand(time(NULL));
37. int array[MAX];
38.
39. SAMPLE;
40. for(int i = 0; i < MAX; ++i)
41. array[i] = rand();
42. selectionSort(array, MAX);
43. SAMPLE;
44.
45. return 0;
46. }
```

ПРИЛОЖЕНИЕ Д

Код модифицированной LR.C

```
1. #include <stdio.h>
2. #include <time.h>
3. #include <stdlib.h>
4. #include “Sampler.h”
5.
6. #define MAX 4859
7.
8. void selectionSort(int array[], int size) {
9. SAMPLE;
10.  for (int i = 0; i < size - 1; i++) {
11.  SAMPLE;
12.  int maxIndex = i;
13.  SAMPLE;
14.  for (int j = i + 1; j < size; j++) {
15.  SAMPLE;
16.  if (array[j] > array[maxIndex]){
17.  SAMPLE;
18.  maxIndex = j;
19.  SAMPLE;
20.  }
21.  SAMPLE;
22.  }
23.  SAMPLE;
24.  int tmp = array[i];
25.  SAMPLE;
26.  array[i] = array[maxIndex];
27.  SAMPLE;
28.  array[maxIndex] = tmp;
29.  SAMPLE;
30.  }
31.  }
32.
33.  int main()
34.  {
35.  SAMPLE;
36.  srand(time(NULL));
37.  int array[MAX], n, c, d, position, t;
38.
39.  SAMPLE;
40.  for(int i = 0; i < MAX ; ++i)
41.  array[i] = rand();
42.  selectionSort(array, CURR);
43.  SAMPLE;
44.
45.  return 0;
46.  }
```