

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. 7304

Пэтайчук Н.Г.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучение возможности измерения динамических характеристик программ с помощью профилировщиков на примере профилировщика SAMPLER.

Постановка задачи.

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.
2. 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:
 - а. измерение только полного времени выполнения программы;
 - б. измерение времен выполнения функциональных участков (ФУ);
3. Выявить "узкие места", связанные с ухудшением производительности программы, ввести в программу усовершенствования и получить новые профили. Объяснить смысл введенных модификаций программ.

Ход выполнения.

1. Была изучена документация монитора SAMPLER, после чего под его управлением были запущены тестовые программы test_cyc.c и test_sub.c. Для проведения измерений использовалась старая версия профилировщика SAMPLER, запуск которого осуществлялся через DOSBox. Результаты работы монитора программ test_cyc.c и test_sub.c продемонстрированы в Таблицах 1 и 2 соответственно:

Исход. Поз.	Прием. Поз.	Общее время (мкс)	Кол-во проходов	Среднее время (мкс)
1 : 9	1 : 11	4335.47	1	4335.47
1 : 11	1 : 13	8668.43	1	8668.43
1 : 13	1 : 15	21678.20	1	21678.20
1 : 15	1 : 17	43348.87	1	43348.87
1 : 17	1 : 20	4335.47	1	4335.47
1 : 20	1 : 23	8670.11	1	8670.11
1 : 23	1 : 26	21672.34	1	21672.34
1 : 26	1 : 29	43348.03	1	43348.03
1 : 29	1 : 35	4334.64	1	4334.64
1 : 35	1 : 41	8670.11	1	8670.11
1 : 41	1 : 47	21676.53	1	21676.53
1 : 47	1 : 51	43348.87	1	43348.87

Таблица 1: Результаты профилирования тестовой программы test_cyc.c

Исход. Поз.	Прием. Поз.	Общее время (мкс)	Кол-во проходов	Среднее время (мкс)
1 : 24	1 : 26	433697.35	1	433697.35
1 : 26	1 : 28	867391.34	1	867391.34
1 : 28	1 : 30	2168480.87	1	2168480.87
1 : 30	1 : 32	4336936.59	1	4336936.59

Таблица 2: Результаты профилирования тестовой программы test_sub.c

- При помощи Borland C++ была скомпилирована программа, написанная на Си, из первой лабораторной работы (program.cpp) после чего была запущена под управлением SAMPLER'a в режиме измерения полного времени выполнения программы. Результаты измерения приведены на Таблице 3:

Исход. Поз.	Прием. Поз.	Общее время (мкс)	Кол-во проходов	Среднее время (мкс)
1 : 37	1 : 50	5521.38	1	5521.38

Таблица 3: Результаты профилирования программы из первой лабораторной работы (полное время работы программы)

3. Программ из первой лабораторной работы была разбита на функциональные участки следующим образом:

a. Функция `main`:

- i. строка 42 – строка 49: начало работы функции, объявление переменных, инициализация генератора псевдослучайных чисел;
- ii. строка 49 – строка 55: цикл по генерации исходных данных;
- iii. строка 55 – строка 57: вызов функции `linfit1`, окончание работы функции;

b. Функция `linfit1`:

- i. строка 7 – строка 15: начало работы функции, объявление переменных;
- ii. строка 15 – строка 26: цикл по вычислению сумм;
- iii. строка 26 – строка 32: вычисление параметров для линеаризации;
- iv. строка 32 – строка 37: цикл для заполнения результирующего массива;

4. Разбитая на функциональные участки программа была скомпилирована и запущена под управлением SAMPLER'a. Результаты профилирования показаны на Таблице 4:

Исход. Поз.	Прием. Поз.	Общее время (мкс)	Кол-во проходов	Среднее время (мкс)
1 : 7	1 : 15	191.09	1	191.09
1 : 15	1 : 26	2028.19	1	2028.19
1 : 26	1 : 32	590.02	1	590.02
1 : 32	1 : 37	751.77	1	751.77
1 : 37	1 : 57	1.68	1	1.68
1 : 42	1 : 49	254.78	1	254.78
1 : 49	1 : 55	1727.32	1	1727.32
1 : 55	1 : 7	5.87	1	5.87

Таблица 4: Результаты профилирования программы из первой лабораторной работы (разбитие на функциональные участки)

Суммарное время работы $T = 5550,72$ мкс.

По результатам профилирования видно, что наибольшее время выполнения у функционального участка с циклом вычисления сумм, в то же время выполнения цикла можно уменьшить за счёт исключения вычисления суммы `sum_y2`, которая используется только для вычисления `suu`, которая нигде больше не используется, соответственно можно так же удалить переменную `suu`. Также можно заменить `xi` и `yi` на `x[i]` и `y[i]` соответственно в вычислениях сумм `sum_x`, `sum_y`, `sum_x2` и `sum_xy` и исключить эти переменные. Таким образом, должно уменьшится время выполнения участков 1 : 7 – 1 : 15, 1 : 15 – 1 : 26 и 1 : 26 – 1 : 32. Изменённая программа была записана в файл `program_update.cpp`.

5. Изменённая программа была скомпилирована и запущена под управлением SAMPLER'a. Результаты профилирования показаны на Таблице 5:

Исход. Поз.	Прием. Поз.	Общее время (мкс)	Кол-во проходов	Среднее время (мкс)
1 : 7	1 : 14	155.89	1	155.89
1 : 14	1 : 22	1789.34	1	1789.34
1 : 22	1 : 27	470.17	1	470.17
1 : 27	1 : 32	698.13	1	698.13
1 : 32	1 : 52	1.68	1	1.68
1 : 37	1 : 44	254.78	1	254.78
1 : 44	1 : 50	1727.32	1	1727.32
1 : 55	1 : 7	5.87	1	5.87

Таблица 5: Результаты профилирования изменённой программы из первой лабораторной работы

Суммарное время работы $T = 5103,18$ мкс, уменьшение времени работы составило 447,54 мкс (~ 8%).

Выводы.

В ходе выполнения лабораторной работы была изучена возможность измерения динамических характеристик программ с помощью профилировщиков и было измерено с помощью профилировщика SAMPLER время выполнения всего кода и время выполнения функциональных участков тестовых программ test_cyc.c и test_sub.c, а также программы из первой лабораторной работы.

В ходе профилирования было выяснено, что на цикл с вычислением сумм приходится наибольшее время выполнения среди всех функциональных участков, после чего была проведена оптимизация программы за счёт удаления ненужных переменных и вследствие этого сокращения ненужных вычислений, что привело к уменьшению времени работы на 447,54 мкс, то есть на 8% от времени работы неоптимизированной программы.