

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Качество и метрология программного обеспечения»
Тема: Измерение характеристик динамической сложности программ
с помощью профилировщика SAMPLER_v2
Вариант №5

Студент гр. 8304

Ивченко А.А.

Преподаватель

Кирияничков В.А.

Санкт-Петербург

2022

Цель работы.

Изучить методы измерения характеристик динамической сложности программ на примере программы SAMPLER_v2

Ход выполнения работы.

1. Из исходных кодов были собраны тестовые программы и инструменты.

2. Под управлением SAMPLER была выполнена программа test_cyc.c. При этом для sampler-repeat использовались параметры 1000 запусков и 5 пропусков.

исх	прием	общее время	кол-во проходов	среднее время
13	15	2887.600	1	2887.600
15	17	5734.400	1	5734.400
17	19	18018.000	1	18018.000
19	21	32020.300	1	32020.300
21	24	2898.500	1	2898.500
24	27	5683.700	1	5683.700
27	30	14116.600	1	14116.600
30	33	28207.700	1	28207.700
33	39	2861.200	1	2861.200
39	45	5730.000	1	5730.000
45	51	14153.900	1	14153.900
51	57	28261.500	1	28261.500

Рисунок 1. Отчет по программе test_cyc

3. Под управлением SAMPLER была выполнена программа test_sub.c

При этом для sampler-repeat использовались параметры 100 запусков и 5 пропусков.

исх	прием	общее время	кол-во проходов	среднее время
30	32	23195431.000	1	23195431.000
32	34	46331439.400	1	46331439.400
34	36	115860696.000	1	115860696.000
36	38	231491306.200	1	231491306.200

Рисунок 2. Отчет по программе test_sub

Из таблиц видно, что при увеличении числа итераций цикла увеличивается время выполнения программы. При этом видно, что второй проход циклов того же размеры происходит быстрее. Это может быть связано с использованием кэша процессора.

4. Была выполнена под управлением монитора SAMPLER программа BubbleSort из лабораторной работы №1. Результат измерений для полного времени выполнения функции sort представлен на рисунке 3. Исходный код этой программы представлен в файле main1.c.

исх	прием	общее время	кол-во проходов	среднее время
35	37	6481.200	1	6481.200

Рисунок 3. Результат работы Sampler для измерения полного времени выполнения функции sort

Было выполнено разбиение программы BubbleSort на функциональные участки. Исходный код программы BubbleSort, разбитый на функциональные участки, представлен в файле main2.c. Полученные с помощью программы SAMPLER результаты представлены на рисунке 4.

исх	прием	общее время	кол-во проходов	среднее время
15	17	4.200	1	4.200
17	19	23.900	1	23.900
19	21	115.300	20	5.765
21	24	-1393.300	20	-69.665
24	27	1619.400	151	10.725
24	33	2338.700	289	8.092
27	29	1693.400	151	11.215
29	31	742.600	151	4.918
31	33	691.000	151	4.576
33	24	5060.900	420	12.050
33	35	329.100	20	16.455
35	19	147.200	19	7.747
35	37	15.500	1	15.500

Рисунок 4. Результат Sampler для функции sort, разбитой на участки

В итоге общее время выполнения – 11387.9 мкс.

Большое количество времени тратится на вызов функции swar, в которой происходит обмен местами двух элементов массива. Для

улучшения данной функции обмен из вспомогательной функции был перенесен в основную. Модифицированный код программы представлен файле main3.c.

исх	прием	общее время	кол-во проходов	среднее время
8	10	6.700	1	6.700
10	12	18.000	1	18.000
12	14	111.300	20	5.565
14	17	159.400	20	7.970
17	20	-132.300	151	-0.876
17	28	2191.800	289	7.584
20	24	1257.200	151	8.326
24	26	766.500	151	5.076
26	28	772.200	151	5.114
28	17	2454.600	420	5.844
28	30	313.200	20	15.660
30	12	184.000	19	9.684
30	32	21.500	1	21.500

Рисунок 5. Результат Sampler для модифицированной функции sort.

В итоге общее время выполнения после модификации – 8124.1 мкс.

Выводы.

В ходе выполнения лабораторной работы были изучены возможности измерения динамических характеристик программ с помощью профилировщиков на примере профилировщика SAMPLER. Для программы, взятой из первой лабораторной работы, было выполнено измерение времени работы, с последующим выявлением узкого места и его устранения – в результате чего удалось добиться более быстрого выполнения функции sort.