

Федеральное агентство связи
Сибирский государственный университет телекоммуникаций и информатики

Лабораторная работа №2а

Выполнил: студент группы ИП-211
Оганесян Альберт
Лацук Андрей
Проверил:
Профессор кафедры ПМиК
Малков Е. А.

Новосибирск 2024

Цель: знакомство с инструментом профилирования программ на платформе **Linux**.

Инструментарий: редактор **vim**, компилятор **gcc**, профилировщик **gprof**. Язык **C**.

Ход работы:

1. Напишем программу с реализацией функции вычисления скалярного произведения двух векторов и добавим функционал для подсчета времени. (Рис 1.1)

```
start = clock(); // Начало отсчета времени
Init(vector1, size, 0);
Init(vector2, size, 1);
InnerProduct(vector1, vector2, size);
end = clock(); // Конец отсчета времени
cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
```

Рис. 1.1 логика подсчета времени

Соберем исполняемый файл с флагом `-pg` и запустим его (Рис 1.2)

```
PS C:\Users\User\Documents\GitHub\OS\2a\Vector> gcc -pg -Wall VectorProduct.c -o VectorProduct
PS C:\Users\User\Documents\GitHub\OS\2a\Vector> ./VectorProduct
SIZE = 20000000 Runtime = 0.194000
```

Рис. 1.2 Сборка и вывод программы

Теперь получим информацию от профилировщика при помощи команды `gprof` (Рис. 1.3)

```
albert@DESKTOP-700AJI4:/mnt/c/Users/User/Documents/GitHub/OS/2a/Vector$ gprof ./VectorProduct
Flat profile:

Each sample counts as 0.01 seconds.
 %   cumulative   self           self      total
time  seconds    seconds   calls   ms/call  ms/call  name
 75.00      0.15      0.15         2     75.00    75.00  Init
 25.00      0.20      0.05         1     50.00    50.00  InnerProduct
```

Рис. 1.3 Получение и вывод информации о времени

2. Занесем данные в таблицу для сравнения

Количество элементов	Время при помощи clock()	Время профилировщика
1000000	0.010002	0.01

2000000	0.010001	0.01
3000000	0.025264	0.02
4000000	0.038756	0.03
5000000	0.042600	0.04
6000000	0.062406	0.06
7000000	0.062153	0.06
8000000	0.075585	0.07
9000000	0.087640	0.08
10000000	0.102628	0.10
11000000	0.106239	0.10
12000000	0.130001	0.13
13000000	0.137554	0.13
14000000	0.135705	0.13
15000000	0.141797	0.14
16000000	0.166481	0.16
17000000	0.177291	0.17
18000000	0.203878	0.20
19000000	0.192385	0.19
20000000	0.212902	0.21

Вывод: Мы познакомились с инструментом профилирования программ gprof и сравнили измерения времени выполнения программы с таймером из библиотеки time.h, по результатам измерений в среднем профилировщик показывает меньшее время, что может свидетельствовать о большей точности работы.