Федеральное агентство связи
Сибирский государственный университет телекоммуникаций и информатики

Лабораторная работа times

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

Новосибирск 2024

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

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

Ход работы:

1. Напишем программу с реализацией функции вычисления скалярного произведения двух векторов и добавим функционал для подсчета времени с times. (Рис 1.1) (Рис 1.2) (Рис 1.3) (Рис 1.4)

```
struct tms start_time, end_time;
clock_t start, end;
```

Рис. 1.1 инициализация структур для хранения данных о времени

```
// начала отсчета
start = times(&start_time);

int size = 20 * 1e6;

double *vector1 = (double *)(calloc(size, sizeof(double)));
double *vector2 = (double *)(calloc(size, sizeof(double)));

Init(vector1, size, 0);
Init(vector2, size, 1);

InnerProduct(vector1, vector2, size);

// конец отсчета
end = times(&end_time);
```

Рис. 1.2 фиксация времени до и после выполнения задачи

```
// конвертируем время из тиков в секунды
double total_time = (double)(end - start) / sysconf(_SC_CLK_TCK);
```

Рис. 1.3 получение тиков в секунды при помощи sysconf и конвертация времени

```
 (double)(end_time.tms_stime - start_time.tms_stime) / sysconf(_SC_CLK_TCK));
унд\n", (double)(end_time.tms_utime - start_time.tms_utime) / sysconf(_SC_CLK_TCK));
```

Рис. 1.4 получение stime и utime из объявленных ранее структур

2. Соберем исполняемый файл и запустим его (Рис 2.1)

```
albert@DESKTOP-70OAJI4:/mnt/c/Users/User/Documents/GitHub/OS/times$ ./a.out
Время работы программы:  0.230000 секунд
Системное время:         0.060000 секунд
Пользовательское время:  0.170000 секунд
```

Рис. 2.1 Вывод программы

Время измеренное при помощи сопоставимо с временем, измеренным при помощи gpof (0.21 секунд)

Теперь поменяем точки измерения (уберем саму функцию получения скалярного произведения векторов из расчета времени) (Рис. 2.2) и посмотрим, как изменится соотношение системного времени к пользовательскому. (Рис. 2.3)

```
// начала отсчета
start = times(&start_time);

int size = 20 * 1e6;

double *vector1 = (double *)(calloc(size, sizeof(double)));
double *vector2 = (double *)(calloc(size, sizeof(double)));

Init(vector1, size, 0);
Init(vector2, size, 0);
// конец отсчета
end = times(&end_time);      danyowoj, 3 дня назад • Added lab 'ti
InnerProduct(vector1, vector2, size);
```

Рис. 2.2 Новое положение точек отсчета

```
albert@DESKTOP-70OAJI4:/mnt/c/Users/User/Documents/GitHub/OS/times$ ./a.out
Время работы программы:  0.200000 секунд
Системное время:         0.060000 секунд
Пользовательское время:  0.140000 секунд
```

Рис. 2.3 Вывод полученного времени

Можем заметить, что системное время не поменялось, а пользовательское время стало меньше.

Вывод: Мы познакомили с инструментом профилирования программ times и сравнили измерения времени выполнения программы профилировщиком gprof по результатам сравнения, можно заметить, что times показывает то же

время, что и gprof, но также передает данные о системном и пользовательском времени, что дает нам более глубокое представление о времени работы программы.