###### МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

###### НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

###### Факультет информационных технологий

**Кафедра параллельных вычислений**

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

«Определение времени работы прикладных программ»

студента 2 курса, 21211 группы

**Великасова Александра Петровича**

Направление 09.03.01 – «Информатика и вычислительная техника»

Преподаватель:

А.Ю.Кудинов

Новосибирск 2022

**Цели работы**

1. Изучение методики измерения времени работы подпрограммы.
2. Изучение приемов повышения точности измерения времени работы подпрограммы.
3. Изучение способов измерения времени работы подпрограммы.
4. Измерение времени работы подпрограммы в прикладной программе

**ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ**

1. Написать программу на языке C или C++, которая реализует выбранный алгоритм из задания.
2. Проверить правильность работы программы на нескольких тестовых наборах входных данных.
3. Выбрать значение параметра N таким, чтобы время работы программы было порядка 15 секунд.
4. По приведенной методике определить время работы подпрограммы тестовой программы с относительной погрешностью не более 1%.
5. Составить отчет по лабораторной работе.
6. Вариант задания №6

**Описание методики для определения времени работы программы.**

1. **Утилита time:**

Измеряет время работы приложения во многих конфигурациях ОС GNU Linux/UNIX.

Выдаёт следующие временные характеристики работы программы:

real – общее время работы программы согласно системному таймеру

user – время, которое работал пользовательский процесс (кроме времени работы других процессов)

sys – время, затраченное на выполнение системных вызовов программы.

Точность: определяется точностью системного таймера и точностью измерения времени работы процесса (см. описание соответствующих таймеров ниже).

Достоинство: готовая утилита, не требуется вносить изменения в программу.

Недостаток: измеряется только время работы всей программы, нет возможности измерить время работы отдельных её частей.

Применение: time ./program.exe

1. **Библиотечная функция clock\_gettime:**

Получает значения системного таймера в ОС Linux/UNIX

С параметром CLOCK\_MONOTONIC\_RAW сохраняет значение системного таймера в структуру struct timespec. Структура состоит из двух полей: tv\_sec и tv\_nsec (можно считать их тип long int), задающих количество секунд и наносекунд (10−9 cек.), прошедших с некоторого неспецифицированного момента времени в прошлом. Разница показаний преобразуется в секунды и выводится на экран.

Кроме системного таймера, функция позволяет получать значения и других таймеров, например времени процесса или потока. Подробнее об этом можно прочитать в документации к этой функции.

Реализация функции clock\_gettime находится в библиотеке rt, поэтому при компиляции программы необходимо добавить ключ компиляции ‘-lrt’.

Точность: зависит от точности системного таймера. Обычно в ОС Windows: 55 мс (55∙10−3 с), в ОС GNU Linux/UNIX: 1 нс (1∙10−9 с).

Достоинство: переносимость – вне зависимости от аппаратного обеспечения функция доступна пользователю, т.к. реализуется ОС.

Недостатки: относительно низкая точность (обычно ниже, чем у счётчика тактов, но выше, чем у функции times), и измеренный интервал включает время работы других процессов, которые работали на процессоре в измеряемый период.

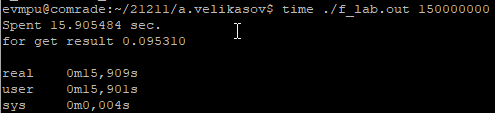
Применение: gcc program.c -o program -lrt**Результат измерения времени работы программы**

1. Написание программы (см. **приложение 1**).
2. Подбор подходящего параметра N, такого что время работы программы находится в интервале 15-30 секунд, подходящим оказался параметр 150.000.000 (Подбирался используя 2 способ подсчета времени).

Изображение выглядит как текст

Автоматически созданное описание

1. После был выполнен подсчет времени первым способом при том же параметре N.



Где:

real – общее время работы программы согласно системному таймеру

user – время, которое работал пользовательский процесс (кроме времени работы других процессов)

sys – время, затраченное на выполнение системных вызовов программы.

**Выводы**

В процессе работы были изучены разные способы измерения времени работы программы и подпрограммы в частности, их достоинства и недостатки.

Была написана программа на языке C, в которой были применены два способа замера времени:

* clock\_gettime – для измерения времени работы части программы.
* time – дает детальную информацию, но замеряет всю программу.

**Приложение 1 (Листинг кода программы)**

