

Задание №4 в рамках вычислительного практикума.
Исследование характеристик программного обеспечения
Ломовской И. В., Кострицкий А. С.

Москва, 2024 г.

На основе задачи №4 ЛР№2 (сортировка) по курсу «Программирование на Си» проведите сравнение производительности работы программы для разных способов работы с элементами одномерного массива:

- использование операции индексации $a[i]$;
- формальная замена операции индексации на выражение $*(a + i)$;
- использование указателей для работы с массивом.

Измерение производительности необходимо выполнить двумя способами:

1. Реализовать инфраструктуру измерения времени выполнения функции в самой программе.
2. Реализовать инфраструктуру измерения времени выполнения функции вне программы. В этом случае внутри программы выполняется замер одного выполнения функции (время выполнения функции выводится на экран или в файл), а повторные замеры выполняются путем многократного запуска самой программы.

Независимо от способа организации замеров повторные замеры продолжаются до тех пор, пока величина RSE превышает 1%. Помимо графиков результаты должны быть представлены в следующих таблицах

Размер	t, мс	Кол-во повторов	RSE, %

Числа в таблица выровнены по правому краю. У вещественных чисел выводятся два знака после запятой.

Для замера времени выполнения функции использовать

- функцию `clock_gettime`, для `clockid` равного `CLOCK_MONOTONIC_RAW`;
- TSC (единица измерений — `ticks`, не нужно переводить в секунды, это не так-то просто).

При выполнении замеров вторым способом использовать только функцию `clock_gettime`.

Уровень оптимизации при сборке программы — O0.

Для сравнения производительности следует реализовать несколько скриптов:

1. `build_apps.sh`, вызвав который, можно получить весь набор необходимых исполняемых файлов.
2. `update_data.sh`, вызвав который, можно добавить некоторые данные в датасет данных исследования.
3. `make_preproc.sh|py`, вызвав который, можно подготовить данные из набора, провести первичный анализ: посчитать среднее арифметическое, медианное, найти максимум и минимум, вычислить нижний и верхний квартили, etc.
4. `make_postproc.sh|py`, вызвав который, можно получить указанные ниже графики.
5. `go.sh`, вызвав который, можно получить данные исследования (скрипт вызывает по очереди предыдущие четыре).

В отчёте привести следующие графики:

1. Обычный кусочно-линейный график зависимости времени выполнения в любых единицах измерения времени от числа элементов массива.
3. Кусочно-линейный график с ошибкой (среднее, максимум, минимум).
4. График с усами (среднее, максимум, минимум; нижний, средний и верхний квартили) для варианта обработки «через квадратные скобки» при уровне оптимизации.

Если Вы проводили исследование с большим количеством вариантов длины массива (получилось много строк в таблицах), достаточно оставить в таблицах десять строк с данными.

Примечания:

1. Помните, что для уменьшения влияния побочных эффектов время замеряют только у целевого алгоритма.
2. Проводить исследование следует со массивами размером от минимального (в некоторых задачах это 1, в некоторых — несколько больше) до, как минимум, 10000. Если производительность Вашего ПК позволяет собрать результаты 20 тестов в час для массивов большего размера, можно поместить эти результаты в

отчёт. Если производительность ПК не позволяет на 10000 элементов собирать хотя бы 20 результатов в час, то максимальный размер обсуждается отдельно.

3. Шаг измерений – 500. Можно плотнее.

4. Помните, что для чистоты исследования состояние системы не должно зависеть от того, какой конкретно алгоритм был использован. Финальное измерение времени всегда проводится тогда, когда система приведена к ожидаемому состоянию.

5. Для отрисовки графиков можно использовать `gnuplot`, `matplotlib` или любой другой «лёгкий» пакет. Два требования: возможность нарисовать всё, что требуется в задании, и наличие консольного интерфейса.

6. При реализации скриптов на Питоне дополнительные пакеты разрешается использовать только в случае, когда стандартной библиотеки недостаточно. Все зависимости в этом случае следует указать. **Базовые функции работы со статистикой в стандартной библиотеке Питона есть.**

7. При желании разрешается добавить другие плоскости исследования: наличие/отсутствие `restrict`, различные максимальный и текущий размеры статического массива, другие флаги оптимизации, etc. Сам смысл задания заключается не только в том, чтобы собрать данные исследования, но и чтобы создать удобный для внесения условий нового исследования интерфейс – поэтому Вы можете проверить, насколько быстро и удобно Вам удалось добавить новое исследование в набор. Результаты дополнительных исследований в отчёт не выносятся, но интересные вопросы могут обсуждаться отдельно.