Министерство образования и науки Российской Федерации

Новосибирский национальный исследовательский государственный университет

ЭВМ и периферийные устройства  
Отчет по лабораторной работе № 2

«ИЗУЧЕНИЕ ОПТИМИЗИРУЮЩЕГО КОМПИЛЯТОРА»

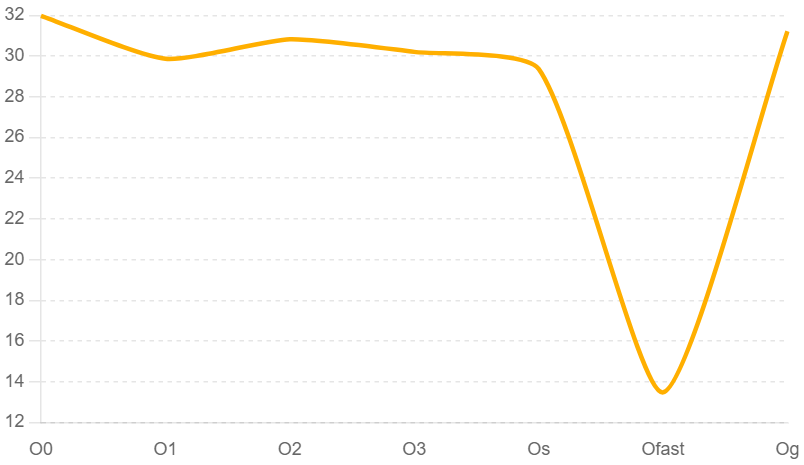
Студент: Данилов Иван Сергеевич  
группа: 23211  
Преподаватель: Мичуров Михаил Антонович

Новосибирск, 2024

**1) Цели лабораторной работы:**

1. Изучение основных функций оптимизирующего компилятора и примеров оптимизирующих преобразований.
2. Изучение различных уровней оптимизации, применяемых компилятором GCC.
3. Получение базовых навыков работы с компилятором GCC.
4. Исследование влияния различных оптимизационных настроек компилятора GCC на время исполнения программ.
5. Ознакомление с примерами оптимизирующих преобразований в GCC и их влиянием на производительность и размер бинарного кода.

**2) Вариант задания №6.**

**3) График зависимости времени от выбранного уровня оптимизации:**

**1) Выводы:**

В ходе лабораторной работы была проведена компиляция и запуск программы с различными уровнями оптимизации компилятора GCC: -O0, -O1, -O2, -O3, -Os, -Ofast и -Og. Целью было измерить время работы программы при каждом уровне оптимизации и проанализировать влияние оптимизаций на производительность.

**Время выполнения программы:**

**-O0 (без оптимизации):** При этом уровне оптимизации не было выполнено никаких оптимизаций кода. Программа работала медленнее всего, что было ожидаемо, ведь генерируется прямой код без каких-либо улучшений.

**-O1:** Включает базовые оптимизации, такие как удаление мертвого кода. Время выполнения программы заметно снизилось по сравнению с **-O0**.

**-O2:** Этот уровень включает более серьезные оптимизации. Прирост производительности был существенным, и время выполнения программы сократилось еще больше.

**-O3:** Включает все оптимизации из **-O2**, а также дополнительные оптимизации направленные на уменьшение времени работы программы. Наблюдалось дополнительное уменьшение времени выполнения, хотя разница между **-O2** и -O3 была незначительной.

**-Os:** Оптимизирует код для минимального размера, что может влиять на производительность. Время выполнения было сопоставимо с **-O2**.

**-Ofast:** Включает все оптимизации из **-O3**, а также оптимизации для математических функции, делая их быстрее, но менее точными. Это привело к наибольшей скорости выполнения, но потенциально с риском для точности вычислений.

**-Og:** Оптимизирует код для отладки. Производительность была лучше, чем при **-O0**, но хуже, чем при остальных уровнях оптимизации.

Результаты лабораторной работы показали, что выбор уровня оптимизации компилятора существенно влияет на производительность программы. Оптимизации могут значительно уменьшить время выполнения, однако следует учитывать возможное влияние на точность вычислений и размер исполняемого файла.