

Тема №4. Выбор оптимальной опции оптимизации

Составитель: Фёдоров С. А.

Редакция от: 18.03.2017

Цель работы — опции оптимизации, оптимальные для вашего приложения.

Часть 1. Выбор подходящего уровня оптимизации

На основе примера, демонстрирующего различные уровни оптимизации, написать первый сценарий, выполняющий следующие действия *в цикле*:

- Компиляцию **вашего приложения, не интерактивно обрабатывающего данные**, на языке C/C++/Fortran/Objective C/Objective C++/Ada с ключами оптимизации:
 - -O0
 - -Os
 - -O1
 - -O2
 - -O3
 - -O2 -march=native
 - -O3 -march=native
 - -O2 -march=native -funroll-loops
 - -O3 -march=native -funroll-loops
- Вычисление времени выполнения программы (time). Приложение без оптимизации должно работать по меньшей мере 20 с.
- Вычисление занимаемого исполняемым файлом дискового пространства (в байтах) (du).

Сценарий должен принимать только имя исходного файла программы.

Вывод сценария должен содержать следующую информацию:

- Текущие опции оптимизации.
- Время затраченное программой на выполнение.
- Занимаемое программой дисковое пространство.

Часть 2. Выбор системного метода оптимизации

Выберите вариант оптимизации, дающий наибольшую производительность для вашего приложения (оптимальная опция).

Проведите оптимизацию с оптимальной опцией, межпроцедурной оптимизацией (см. серию опций -fipa-*) и оптимизацией времени компоновки (-flto). Определите время работы приложения.

Проведите оптимизацию с оптимальной опцией и с оптимизацией с обратной связью (-fprofile-generate/-fprofile-use). Определите время работы приложения.

Проведите оптимизацию с оптимальной опцией, межпроцедурной оптимизацией, оптимизацией времени компоновки и с оптимизацией с обратной связью. Опре-

делите время работы приложения.

Отчётность

Подготовить в электронной форме документ (не отчёт), содержащий таблицу со всеми используемыми вариантами оптимизации и временем работы при них. Сделать вывод по оптимальной стратегии оптимизации вашего приложения.