УДК 004.4'422

Анализ указателей, основанный на правилах перекрытия объектов в памяти.

А.Л. Маркин[1], А.В. Ермолицкий[1]

[1] ЗАО «МЦСТ»

Анализ указателей является важной частью любого оптимизирующего компилятора. Он позволяет вычислять пересечения указателей в памяти и определять, могут ли ссылаться два указателя на один и тот же участок памяти. Эта информация необходима для полноценной работы всех оптимизаций, переставляющих операции работы с памятью [1].

Анализ указателей является технически сложным и ресурсоёмким процессом, а наибольшего эффекта достигает при полном анализе программы. Но на данный момент технологии компиляции в режиме «вся программа» и оптимизации времени связывания ещё не достаточно развиты для повсеместного использования, а стандартом является компиляция в помодульном режиме. В этом режиме наиболее эффективные анализы указателей или не работают, или не дают значимых результатов [2].

Для решения данной проблемы в стандарт языков C и C++ были введены правила перекрытия объектов в памяти, заключающиеся в том, что доступ к объектам может осуществляться только через выражения, имеющие тип, совместимый с этими объектами [3]. Данное правило позволило внедрить в оптимизирующий компилятор специальный анализ указателей, основанный на информации о типах, называемый strict-aliasing. Этот анализ позволяет определить независимость двух обращений в память, если они имеют несовместимые типы.

В работе исследовалось влияние анализа указателей, основанного на правилах перекрытия объектов в памяти, на производительность и надёжность программного обеспечения. Был реализован точный анализ нарушений данного правила, позволяющий выявлять ошибки пользователя, а также потенциально опасные конструкции в программе. Производились замеры потребляемых компилятором ресурсов, а также производительность получаемого когда с включённым анализом и без него в режимах помодульной сборки и сборки в режиме “вся программа”.

Исследование производительности проводилось на наборе бенчмарков SPEC-2000 [4], на котором использование данного анализа позволило ускорить исполнение программ до 8% в помодульном режиме и до 2% в режиме «вся программа».

Исследование надёжности проводилось на основе 387 пакетов GNU окружения ОС «Эльбрус», и показало наличие пакетов с нарушениями правил перекрытия объектов в памяти. Был выявлен 41 пакет с явными нарушениями, а также 157 пакетов с потенциально возможными нарушениями. Обычно такие пакеты собираются с опцией -fno-strict-aliasing,

устраняющей опасность нарушения данного правила, однако реализация точного анализа позволила оставить опцию включённой, точечно не применяя анализ к блокам кода с выявленными нарушениями.

Литература

1. *Allen R., Kennedy K.* Optimizing compilers for modern architectures. Morgan Kaufman. San Francisco, 2002.
2. *Дроздов А.Ю, Владиславлев В.Е*. Межпроцедурный анализ указателей. Информационные технологии, Приложение № 2, 2005.
3. Programming Language C. ISO/IEC 9899:TC2. 6.5 Expressions. - 2005
4. http://www.spec.org/ [Электронный ресурс]