

Смарт-компилятор - помощник программисту

Козлов В.С., ФГБОУ ВО «МГТУ им. Г.И. Носова»/Проектная школа, г. Магнитогорск, РФ
Мазнин Д.Н. - доц. каф. ИиИБ, ФГБОУ ВО «МГТУ им. Г.И. Носова», г. Магнитогорск, РФ

Существует множество языков программирования - интерпретируемых и компилируемых. Компилируемые позволяют писать высокопроизводительные программы, дают возможность взаимодействовать с операционной системой. Интерпретируемые упрощают жизнь программиста путём представления информативных ошибок и ряда абстракций, но не работают на «низком» уровне.

Целью работы является облегчение жизни разработчиков - язык программирования и компилятор должны помогать программисту. При создании, мы должны учитывать, что не существует «бесплатных» абстракций, а компилятор должен увеличивать безопасность и производительность. Главным критерием оптимизации в современных компьютерах является многопоточность исполнения. Этим преимуществом пользоваться трудно, поэтому необходим смарт-компилятор, который бы предугадывал исполнение кода.

Наша задача - создать данное средство разработки. Главным решением является отказ от заголовочных файлов и файлов ресурсов. Это позволит компилятору иметь полный контроль над сборкой. Смарт-компилятор сможет определять неопределённое поведение, лучше отслеживать время жизни объектов, повышать безопасность, а самое главное - обезопасить многопоточное исполнение. Не менее важным будет опциональность всех атрибутов, без последствий со стороны производительности. Компилятор сможет автоматически выставлять константы, атрибуты функций, заменять вызов функции на их тело, выполнять предварительные расчёты.

Синтаксис языка будет C-подобным, так как это определённый стандарт в отрасли. Большинство конструкций будут взяты из C++, но изменены под особенности нашего компилятора. В C++ нас не устраивает большое число атрибутов, заголовочные файлы, не позволяющие компилятору видеть всю картину происходящего, потому мы и отказываемся от этого. Также мы сможем иметь более высокоуровневые абстракции, так как их недостатки будут скомпенсированы средствами разработки.

Результатом работы будет являться язык программирования и компилятор, который будет сочетать преимущества компилируемых и интерпретируемых языков. Благодаря особенностям, язык сможет найти применение в программировании большого числа мобильных устройств.

Список литературы

1. Dick Grune, Kees van Reeuwijk, Henri E. Bal Ceriel J.H. Jacobs, Koen Langendoen Modern Compiler Design Second Edition
2. Ахо А. и др. Компиляторы. Принципы, технологии и инструментарий.
3. Серебряков В.А., Галочкин М.П. Основы конструирования компиляторов.