

Предисловие

Изучение компиляторов является центральным и одним из наиболее востребованных аспектов компьютерных наук. Написание компилятора требует знания исходного языка и целевой машины и обеспечения их взаимосвязи. Наличие современного инструментального обеспечения освобождает программиста от многих утомительных, подверженных ошибкам моментов при создании компилятора.

Мое первое знакомство с компиляторами произошло в 1967 году на лекциях создателя ранних моделей компиляторов для языка ALGOL 60 Питера Наура (Peter Naur) в Университете Ньюкасла. Позднее, в 1974 году, мне посчастливилось прослушать расширенный курс по конструированию компиляторов в Техническом университете Мюнхена (отдел Ф. Л. Бауэра (F. L. Bauer)). С тех пор я, с некоторыми перерывами, работаю с компиляторами и инструментами для их разработки.

Большинство теоретических разработок и методов для конструирования компиляторов возникло в 1970-х, а некоторые из них даже раньше. В то же время развитие языков программирования постоянно ставило перед создателями компиляторов новые задачи: от определения языка ALGOL 60 (ранние работы Наура в Дании, а также Ренделла (Randell) и Рассела (Russell) в Великобритании) до удовлетворения потребностей возникших позже языка Ada и языков объектно-ориентированного программирования. Современные требования Java к сетевой безопасности, эффективности и переносимости вновь ставят перед создателями компиляторов новые проблемы. В то же время появление Java Virtual Machine дает великолепный пример промежуточного языка, позволяющего проиллюстрировать различные аспекты генерации кода и распределения памяти.

Процесс создания компилятора соединяет в себе как творческую, так и рутинную работу. Он требует хорошей инструментальной поддержки, что отчетливо видно при изучении истории развития компиляторов. В наше время доступно множество инструментальных средств для создания компиляторов. Некоторые из них можно получить бесплатно через Internet, но на данный момент наиболее используемыми и распространенными инструментальными средствами являются Unix Lex и YACC (сейчас они могут применяться и на других платформах). В Internet доступны также бесплатные версии Lex и YACC, известные как Flex и Bison. Также можно получить существующие версии turbo Pascal (изначально основанные на C).

Во многих книгах по компиляторам описываются средства Lex и YACC с примерами их использования, но лишь в немногих из них содержится достаточно информации для того, чтобы читатель сам смог воспользоваться этими инструментами для решения своих задач. Именно поэтому одной из задач, поставленных при

написании данной книги, является всестороннее (хотя и не полное, все-таки книга небольшая) рассмотрение Lex и YACC. При этом их использование иллюстрируется не только на примерах, связанных с созданием компиляторов, но и на примерах, затрагивающих другие синтаксически управляемые средства, например, инструменты по вычислению метрик исходного кода.

Основным языком при написании различных алгоритмов на базе Lex и YACC является C. Данная книга прежде всего посвящена компиляции императивных языков, поэтому язык C также будет применяться в качестве исходного языка во многих примерах, описывающих различные аспекты компиляции. В то же время многие свойства языка, компиляцию которого мы желаем рассмотреть, не связаны с C, поэтому в таких случаях будут использоваться другие, более подходящие языки — FORTRAN, Pascal, Ada, ALGOL 68, C++, Java.

Первая часть книги посвящена этапу анализа процесса компиляции. После введения в процесс компиляции (глава 1) следуют главы по определению языка, лексическому анализу, нисходящему и восходящему синтаксическому анализу и семантическому анализу (главы 2–6). Более короткая вторая часть книги посвящена этапу синтеза процесса компиляции. Она содержит две главы, в которых рассмотрено распределение памяти и генерация кода (главы 7–8). Каждая глава завершается серией упражнений, ответы на которые приведены в конце книги. Кроме того, в конце каждой главы помещен список литературы, рекомендуемой для дальнейшего изучения. Значение большинства технических терминов, используемых в книге, можно найти в глоссарии.

Данная книга рассчитана на односеместровый курс по компиляторам и их инструментальным средствам, подобный читаемому в Университете Стратклайда (University of Strathclyde). Названный университетский курс также включает практические занятия, на которых студенты закрепляют навыки по использованию Lex и YACC. Как и другие книги серии “Основы вычислительных систем”, эта книга стремится подать все необходимые аспекты процесса компиляции, не пытаясь охватить всю область.

Благодарности

С особым удовольствием я выражаю свою признательность издательству Prentice Hall за помощь и поддержку в создании настоящей книги. Необходимо также поблагодарить Бориса Когана (Boris Cogan) и Тамару Матвееву (Tamara Matveeva) за их полезные комментарии к тексту. Благодарю свою жену Кейт (Kate) за ее поддержку и терпение, а также за проверку всей рукописи, которую она охотно выполнила. Благодарю рецензентов издательства и редактора серии Рея Велланда (Rey Welland) за их конструктивные замечания на разных этапах создания книги.

Робин Хантер (Robin Hunter)

Глазго

Май 1998