

Л и т е р а т у р а

1. Баррон Д. Ассемблеры и загрузчики. - М.: Мир, 1974.

2. Калечиц В.Е., Лободин Н.И., Нестерова О.И. Кросс-система микроЭВМ "Электроника-60" в ДЭС ЕС ЭВМ. Кросс-компоновщик. Назначение и возможности. - Воронеж: ЦНТИ, ИЛ №10-82 НТД, 1982.

УДК 681.3.06

Н.И.Л о б о д и н

РЕАЛИЗАЦИЯ КРОСС-АССЕМБЛЕРА МИКРОЭВМ
"ЭЛЕКТРОНИКА-60" НА ЕС ЭВМ

(г.В о р о н е ж)

Рассматриваемая реализация кросс-ассемблера ориентирована на выполнение следующих функций:

трансляция программ на языке ассемблера кросс-системы;

генерация объектного кода;

генерация абсолютного кода без промежуточного действия компоновки с целью последующей эмуляции;

управление входными потоками;

управление режимами ассемблирования;

выдача листинга трансляции;

выдача диагностических сообщений об ошибках;

ввод исходных текстов с произвольных носителей ЕС ЭВМ;

использование всех средств ассемблера микроЭВМ;

расширение стандартного языка ассемблера средствами структурного управления.

О с н о в н ы е к о н ц е п ц и и. При создании кросс-ассемблера основное внимание уделялось максимальному использованию всех средств и возможностей большой ЭВМ с целью обеспечения высокоэффективной трансляции. Для удовлетворения этого требования были разработаны нетрадиционные методы трансляции, основным из которых является концепция применения промежуточного текста.

Данный транслятор ассемблирует исходный текст за два просмотра. На первом осуществляется основная обработка всех конструкций языка, вычисляется и формируется все, что можно вычислить и сформировать на этом этапе. В силу этого программы первого просмотра являются доста-

точно сложными, однако такой подход позволяет значительно упростить программы второго просмотра, тем более, что многие его блоки несущественно отличаются от блоков первого просмотра. Такие отличия удалось свести к небольшим настройкам соответствующих блоков, не влияющих на эффективность программы.

Результатом работы первого просмотра является промежуточный текст, состоящий из последовательности специальных директив, выполняемых на втором просмотре. Такой подход позволил избавиться от повторного анализа исходного текста и увеличить скорость работы второго просмотра.

П е р в ы й п р о с м о т р. Как уже отмечалось выше, основная работа по трансляции исходной программы производится ассемблером на первом просмотре. В отличие от многих существующих схем ассемблирования в данной реализации на первом просмотре осуществляется не только распределение памяти и составление таблицы имен, но и проводится вся возможная работа по генерации машинного кода. Исключения составляют действия, которые не могут быть выполнены из-за ссылок вперед. Они откладываются до второго просмотра.

В процессе последовательной переработки операторов исходного текста модули первого просмотра обращаются к сканеру для выборки символов исходного текста. Они получают от сканера информацию о классе очередного выбранного символа (например: метка, мнемоника директивы, операнд, операция и т.д.) и конкретное значение выделенного символа в соответствии с принятыми соглашениями о классах сканируемых символов. В силу того, что принадлежность символа тому или иному классу определяется по контексту, предусмотрено несколько режимов работы сканера.

Обработка очередного оператора начинается с обращения к сканеру в режиме "операция". В зависимости от класса выданного символа производится переключение на соответствующую подпрограмму его обработки. По завершении разбора оператора выполняется проверка на конец и, если текст оператора не кончился, генерируется сообщение об ошибке и осуществляется принудительный выход на конец оператора. Затем управление передается на разбор следующего оператора.

Метки, присутствующие перед оператором, отделяются и по одной записываются в таблицу имен без значений. Проверяется лишь корректность их задания. Ссылки на таблицу имен накапливаются в специальной таблице. Окончательная обработка меток проводится после определения типа соответствующего оператора и выполнения связанных с ним вспомогательных действий, влияющих на значение и характеристики меток.

При обработке оператора прямого присваивания имени запоминается отсылка на таблицу имен, вычисляется выражение в правой части и результат (или отсылка на полиз) заносится в таблицу имен. В случае повторного определения имени в промежуточный текст выдается директива смены значений, кроме того, значение имени по первому определению сохраняется в специальной таблице, что дает возможность восстановить первое определение перед началом второго просмотра.

Для прямого присваивания счетчику адреса вычисляется выражение в правой части и, после проверки соответствия его типа типу секции, присваивается счетчику команд. В промежуточный текст выдается директива изменения счетчика команд.

Суть обработки машинных команд состоит в следующем. В таблице имен вместе с мнемониками команд хранятся так называемые схемы разбора. В силу того, что в командах в различных сочетаниях могут встречаться операнды трех типов - "абсолютное выражение", "смещение", "поле адресации" - обработка операндов каждого типа не зависит от схемы разбора и выделена как самостоятельная единица программы. Если в операндах задано выражение, значение которого нельзя вычислить из-за неопределенности имен, то псевдополиз таких выражений выводится в промежуточный текст, чтобы не перегружать поле псевдополизов. Блок обработки машинных команд начинает работу с выравнивания счетчика команд, затем присваиваются значения меткам, выбирается в поле сборки заготовка команды и производится переключение на соответствующую схему разбора. После этого специализированные подпрограммы производят формирование командного слова и, если необходимо, выдают дополнительные слова в промежуточный текст. В случае невозможности формирования команд выдаются соответствующие директивы промежуточного текста для доформирования команд.

При обработке директивы конца модуля происходит настройка для работы с промежуточным текстом и управление передается программам второго просмотра.

В т о р о й п р о с м о т р. Основной целью второго просмотра является доформирование команд, выдача объектного кода и листинга транслируемой программы. На вход поступает промежуточный текст, полученный на первом просмотре.

Для управления вторым просмотром разработана специальная подпрограмма, осуществляющая дешифрацию кода директивы и передачу управления подчиненным подпрограммам. Она является одной из наиболее часто используемых подпрограмм (во время трансляции работает тысячи раз), поэтому при ее реализации основное внимание уделялось скорости работы. Для

выполнения этого требования были применены всего лишь четыре быстрые команды. Особому рассмотрению подлежат директивы промежуточного текста: ввод "объектного текста", деформирование команд, накопление листинга, управление трансляцией.

Первый класс директив предназначен для вывода машинного кода в файл компоновщика. "Объектный текст" может поступать из промежуточного текста и рабочих полей, в которые заносятся результаты вычисления выражений или доформированные команды.

Работа по доформированию команды начинается с очистки поля для сборки команды, затем вычисляются выражения, переданные в промежуточный текст, и формируются нужные поля команды. Эти действия осуществляются с помощью тех же блоков, что и на первом просмотре (возможно с небольшими перенастройками). Так, например, появление неопределенных имен в ходе вычисления выражения на первом просмотре вызывает формирование директивы вычисления этого выражения, тогда как на втором просмотре появление неопределенного имени приводит к выдаче сообщения об ошибке. При вводе директивы с машинной командой заготовка приформовывается к подготовленным в поле сборки полям и инициируется выполнение директивы ввода сформированной команды.

С помощью директив накопления листинга формируется очередная строка листинга, выдаются сообщения об ошибках и может изменяться поле адреса оператора. При этом оператор может выдаваться с нумерацией, без нее или без адреса оператора.

Директивы управления позволяют управлять выдачей листинга, временно задерживать ввод объектного кода, производить смену буферов промежуточного текста, изменять значение имен, счетчика адреса и тип секции. По директиве конца происходит завершение трансляции и выдача итогового сообщения.

Кросс-ассемблер разработан в рамках операционных систем ДОС и ОС ЕС ЭВМ, все модули написаны на языке ассемблера. Функционирует он в составе кросс-системы программирования для микроЭВМ "Электроника-60" на ЕС ЭВМ.

Изложенные принципы реализации могут быть полезны при разработке других ассемблеров.

Л и т е р а т у р а

1. Донован Д. Системное программирование. - М.: Мир, 1975.
2. Калечиц В.Е., Лободин Н.И., Шукин А.Я. Кросс-система ЭВМ "Электроника-60" на ЕС ЭВМ. - В сб.: Методы трансляции. Ростов: Изд-во Ростовского университета, 1981.