

В.Е.Калечиц, Н.И.Лободин

РЕДАКТИРОВАНИЕ ОБЪЕКТНЫХ МОДУЛЕЙ МИКРОЭВМ

"ЭЛЕКТРОНИКА-60" В КРОСС-СИСТЕМЕ

(г.В о р о н е ж)

В статье обсуждается редактирование связей между объектными модулями, полученными в результате трансляции кросс-ассемблером или конвертированные непосредственно с микроЭВМ "Электроника-60". Основные концепции редактирования излагаются на примере компоновщика кросс-системы микроЭВМ "Электроника-60" на ЕС ЭВМ.

В обязанности рассматриваемого компоновщика входит:

поиск объектных модулей в библиотеке и/или файле кросс-ассемблера;

связывание и перемещение объектных модулей;

формирование загрузочного модуля;

каталогизация загрузочного модуля в спецфайл или библиотеку;

выдача протокола компоновки и диагностических сообщений.

Получаемые компоновщиком загрузочные модули могут в последующем выполняться (после соответствующего конвертирования) на микроЭВМ "Электроника-60" или непосредственно эмулироваться и отлаживаться в кросс-системе на ЕС ЭВМ.

При трансляции типичной является ситуация, когда в обрабатываемых модулях используются обращения к внешним (глобальным) именам. В этих случаях из-за невозможности определения их адресов ассемблер вынужден соответствующие позиции в командах обращений заполнять нулями и формировать необходимые инструкции перемещения для глобальных имен. С целью обеспечения эффективной компоновки ассемблер создает дополнительно таблицы глобальных имен для каждого модуля, содержащие описания глобальных имен из других модулей и внутренних имен, к которым производится обращение извне. Таким образом, смысл связывания состоит в формировании фактических адресов для команд обращений к глобальным именам.

Процесс редактирования может выполняться в одном из двух режимов - явного или автоматического подключения модулей.

В режиме явного объявления модулей компоновщику после ввода очередного модуля становятся известны все его глобальные имена. Неопределенные имена, обнаруженные по завершении обработки всех модулей, рассматриваются как ошибочные.

Во втором режиме для неопределенных имен осуществляется поиск соответствующих модулей в библиотеке, который может быть успешным только при условии совпадения пускового адреса и имени модуля в оглавлении библиотеки. Это может быть нарушено для случая принудительного подключения модуля.

Традиционный редактор связей общего назначения требует довольно много машинного времени, и нередко оказывается, что редактирование происходит дольше, чем компилирование. Обычно процесс редактирования выполняется за два просмотра. При этом для выполнения перемещения объектный текст на первом просмотре копируется в рабочий файл, а на втором — считывается из него.

В данной реализации разработан высокоэффективный алгоритм одно-просмотрового редактирования связей. Существенное повышение скорости достигается за счет отмены промежуточного копирования текста. Для разъяснения работы этого алгоритма целесообразно рассмотреть структуру объектных модулей кросс-системы.

Точно так же, как исходная программа содержит директивы, управляющие режимами ассемблирования, объектная программа может включать кроме оттранслированного текста команды, управляющие работой компоновщика. Объектный модуль может содержать следующую информацию: глобальные элементы; команды перемещения; определение абсолютной секции; определение относительной секции; блоки текста; специальные признаки.

Глобальные элементы могут быть четырех типов: имя модуля; глобальное имя; пусковой адрес (точка входа в программу); относительная секция.

Команды глобального и внутреннего перемещения содержат информацию, необходимую для настройки оттранслированных блоков текста в соответствии с указанным адресом загрузки в памяти.

Как уже отмечалось выше, функции распределения памяти, связывания модулей и перемещения текстов выполняются за один просмотр, что предполагает необходимость сохранения информации о тех командах кросс-компоновщика, которые невозможно выполнить из-за неопределенности используемых в них глобальных имен. Запоминаемая информация о таких командах включает: абсолютный адрес перемещаемого текста; значение перемещаемого текста до поступления невыполнимой команды; длину перемещаемого текста; информацию о глобальном имени, используемом при настройке текста.

Невыполнимые команды вместе с перечисленными данными образуют так называемые "отложенные команды". Сохраняются они в специальной области памяти и выполняются по мере определения глобальных имен.

Возможно, что для одного и того же участка текста перемещение выполняется несколько раз. Такая ситуация возникает при обработке выражений с несколькими глобальными именами. В этом случае группе команд соответствует "цепочка", в которой содержатся сведения о длине перемещаемого текста, коде операции и используемом глобальном имени.

Следует отметить также разумное изменение структуры объектных модулей кросс-системы по сравнению со стандартом "Электроника-60", связанное с тем, что объектные модули микроЭВМ "Электроника-60", во-первых, страдают избыточностью информации о командах компоновщика и, во-вторых, не допускают возможности обработки даже простых выражений из глобальных имен, что весьма существенно при разработке языков высокого уровня для микроЭВМ.

И, наконец, эффективность работы компоновщика повышалась за счет сокращения времени на повторные просмотры оглавления при поиске объектных модулей в библиотеке. Соответствующая таблица ссылок на объектные модули заполняется при первом обращении к библиотеке. Это избавляет от необходимости считать и просматривать оглавление библиотеки, в случае, когда ссылка возникла в результате ошибки и, кроме того, позволяет решать вопрос с зависимыми подпрограммами и внутренними глобальными именами.

Ниже приводится алгоритм реализации кросс-компоновщика на псевдокоде.*

[КРОСС-КОМПОНОВЩИК]

НАЧАЛО

настройка: определение границ программы;
установка указателей таблиц и специальных признаков;
открытие файлов данных; инициализация переменных

ЦИКЛ

ввод очередной записи с внешнего устройства
определение типа элемента

ВЫБОР тип элемента ИЗ

ВАРИАНТ объектный модуль

ОБРАБОТКА ОБЪЕКТНОГО МОДУЛЯ

ВАРИАНТ конец задания

завершение обработки загрузочного модуля

завершение всей работы

* В силу ограниченности объема статьи несущественные детали при описании алгоритма были опущены.

ИНАЧЕ

обработка оператора определения характеристик
загрузочного модуля или оператора принудительного
подключения модуля

КВЫБОР

КЦИКЛ

КОНЕЦ

[ОБРАБОТКА ОБЪЕКТНОГО МОДУЛЯ]

НАЧАЛО

ЦИКЛ

определение типа элемента

ВЫБОР тип элемента ИЗ

ВАРИАНТ глобальный элемент

ОБРАБОТКА ГЛОБАЛЬНОГО ЭЛЕМЕНТА

ВАРИАНТ команда перемещения

ОБРАБОТКА КОМАНДЫ ПЕРЕМЕЩЕНИЯ

ВАРИАНТ команда "LIMIT"

Занесение в таблицу "отложенных команд" информации о
нижней и верхней границах программы. и адресах настройки
блока текста.

ВАРИАНТ секция

установка признака абсолютной или относительной секции

ВАРИАНТ блок текста

выполнение операции по пересылке блока текста
в выходной буфер и вывод буфера при его
заполнении

ВАРИАНТ конец записи

ввод очередной записи модуля

ВАРИАНТ конец модуля

увеличение указателя таблицы модулей и его сохранение

ИНАЧЕ

вывод диагностического сообщения об ошибке

КВЫБОР

КЦИКЛ

КОНЕЦ

[ОБРАБОТКА ГЛОБАЛЬНОГО ЭЛЕМЕНТА]

НАЧАЛО

определение типа элемента

ВЫБОР тип элемента ИЗ

ВАРИАНТ имя модуля

сохранение имени в таблице имен модулей
и установка начальной длины относительной секции

ВАРИАНТ относительная секция

сохранение адреса начала и длины секции, определение
адреса секции следующего модуля

ВАРИАНТ адрес перехода

вычисление и сохранение точки входа в программу

ВАРИАНТ глобальное имя

поиск имени в таблице глобальных имен

ЕСЛИ имени нет

ТО записать в таблицу обнаруженное имя и его
характеристики

ИНАЧЕ

ЕСЛИ имя определено в модуле

ТО отметить, что оно дублируется

КЕСЛИ

КЕСЛИ

КВЫБОР

КОНЕЦ

[ОБРАБОТКА КОМАНДЫ ПЕРЕМЕЩЕНИЯ]

НАЧАЛО

определение типа команды, адреса и размера изменяемого
текста

ВЫБОР тип команды ИЗ

ВАРИАНТ внутреннее перемещение

определение адреса начала относительной секции модуля
и выполнение перемещения текста

ВАРИАНТ глобальное перемещение

поиск глобального имени в таблице имен

ЕСЛИ имя определено

ТО выполнение перемещения текста

ИНАЧЕ

ЕСЛИ "цепочка" невыполненных команд продолжается

ТО запоминание в таблице "отложенных команд" номера
глобального имени, когда операции и длины текста;
корректировка указателей "цепочки"

ИНАЧЕ

запоминание в таблице "отложенных команд"
абсолютного адреса и значения изменяемого
текста, номера глобального имени, кода
операции, адреса последней "цепочки"

КЕСЛИ

КЕСЛИ

КВЫБОР

КОНЕЦ

[ЗАВЕРШЕНИЕ ОБРАБОТКИ ЗАГРУЗОЧНОГО МОДУЛЯ]

НАЧАЛО

ЕСЛИ задан режим автоматического подключения модулей

ТО

обработать недостающие модули

КЕСЛИ

обработка "отложенных команд"

запись в абсолютный текст пускового адреса

обработка конца модуля

печать протокола обработки загрузочного модуля

КОНЕЦ

[ОБРАБОТКА "ОТЛОЖЕННЫХ КОМАНД"]

НАЧАЛО

НИКЛ ПОКА не конец таблицы цепочек

выборка адреса загрузки и значения перемещаемого текста

определение длины перемещаемого текста

НИКЛ ПОКА не конец "цепочки"

определение значения глобального имени

запись в "заплату" (корректируемое место абсолютного
текста) нового значения изменяемого текста

обработка "заплаты" как блока текста

КЦИКЛ

ЦИКЛ

КОНЕЦ

Кросс-компоновщик разработан в операционных системах ДОС и ОС ЕС ЭВМ на языке ассемблера с применением макрокоманд структурного программирования. Изложенные принципы реализации могут быть полезными при создании редакторов связей, ориентированных на специальные виды работ.

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-