

14. Ассемблирование программ

Ассемблирующая программа (АП) является примером простейшего транслятора, переводящего исходный текст на языке ассемблера в двоичное представление. Для достижения этой цели АП должна решить следующие основные задачи:

- назначить адрес размещения в памяти каждому элементу данных и каждой команде
- заменить мнемонические коды операций двоичными эквивалентами
- обработать операндные части команд и перевести их во внутреннее представление, в частности – заменить все символьные операнды соответствующими адресами
- сформировать машинные команды определенного формата
- создать выходной объектный модуль, пригодный для последующей компоновки с другими модулями

Дополнительно АП должна:

- вывести сообщения об обнаруженных при ассемблировании ошибках
- сформировать листинг программы

АП просматривает исходный текст последовательно, обрабатывая строку за строкой. Каждая строка сканируется и из нее выделяются основные составляющие – имена областей памяти и метки команд, мнемокоды операций и имена управляющих директив, операнды и комментарии. Жесткий формат строк ассемблерного текста существенно облегчает этот разбор. Для управления процессом ассемблирования используются директивы или псевдокоманды (DB, DW, PROC, END и др.).

Последовательная обработка строк практически всегда приводит к необходимости решения задачи, известной как “обработка ссылок вперед”. Эта ситуация возникает, когда при обработке очередной команды в ее операндной части встречается неопределенное символьное имя. Имя

считается неопределенным, если с ним не связан соответствующий адрес памяти. Связь имени с адресом выполняется при распознавании этого имени в поле метки, поэтому если определение имени происходит ПОЗЖЕ его использования в команде, АП не может заменить имя соответствующим адресом.

Различают ссылки вперед на данные и на команды. Пример ссылки вперед на данные:

```
MOV AX, MyData
.....
MyData DW 1234
```

Такие ссылки легко устраняются с помощью вполне логичного требования объявления данных ДО их использования в командах. Именно по этой причине в ассемблерных программах сначала объявляется сегмент данных, а уж потом следует кодовый сегмент.

К сожалению, также легко устранить ссылки вперед на команды невозможно, поскольку такая ссылка возникает при передаче управления на команду, расположенную ПОСЛЕ текущей:

```
JMP MyLabel
.....
MyLabel: MOV AX, BX
```

Для решения этой задачи АП можно реализовать одним из двух следующих способов:

- двухпросмотровая (двухпроходная) схема предполагает выполнение двух проходов по исходному тексту, при этом логика каждого прохода очень простая, но чуть возрастают затраты времени
- однопросмотровая (однопроходная) схема реализует только один проход по тексту с более сложной логикой обработки строк и некоторым увеличением затрат памяти.

Поскольку двухпроходная схема проще для объяснения, далее сначала будет рассмотрена именно эта схема, а потом – алгоритмически более сложная однопроходная схема.