

45840

МС-341

11/1x-63

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

11 - 4656

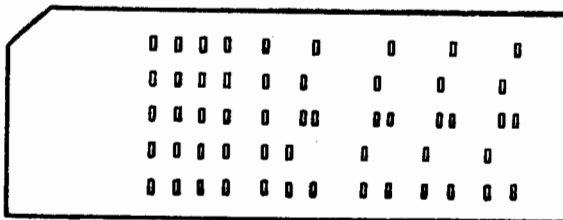


ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ
И АВТОМАТИЗАЦИИ

Б. Жаргал

АССЕМБЛЕР ДЛЯ ЭВМ "МИНСК-22"

1969

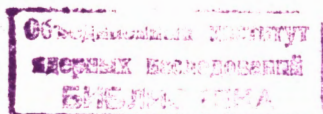


11 - 4656

ОБЪЕДИНЕННЫЙ ИНСТИТУТ
ЯДЕРНЫХ ИССЛЕДОВАНИЙ
ЛВТА

Б. Жаргал

АССЕМБЛЕР ДЛЯ ЭВМ "МИНСК-22"



дн 1/1467

ВВЕДЕНИЕ

"Ассемблер" является транслирующей программой, которая преобразовывает условную символическую программу, написанную по определенным правилам (будем называть её исходной программой) в программу в коде "Минск-22, готовую для работы (рабочая программа).

Применение ассемблера позволяет:

1. Писать исходную программу в символическом виде, не задавая заранее распределения памяти и не вычисляя истинные адреса.

Распределение памяти и присвоение истинных адресов производится автоматически в процессе трансляции.

2. Легко вносить изменения в исходную программу с целью отладки и корректировки.

Кроме трансляции, данный вариант ассемблера может дополнительно:

1. Распечатывать на АЦПУ исходную программу.

2. Выдавать на АЦПУ "Листинг" программы, т.е. рабочую программу с поставленной в соответствие с ней символической программой и дополнительными признаками.

3. Выдавать рабочую программу на ТБПМ.

4. Выдавать на ТБПМ таблицу распределения памяти.

5. Выводить рабочую программу на перфатор № I.

6. Записывать рабочую программу на МЛ.

ВХОДНОЙ ЯЗЫК ПРОГРАММЫ

Символы входного языка.

Буквы: Все буквы латинского алфавита.

Цифры: 0,1,2,3,4,5,6,7.

Символы: (|) | < | > | * | = | ↑

Знаки: + | -

Разделители: : (двосточие)

, (запятая)

/ (косая черта)

. (точка)

Символическая инструкция - это условная запись команд и констант, в которой одна часть от другой отделяется разделителями, а в отдельных частях инструкции возможно употребление всех символов входного языка (кроме разделителей).

Применение разделителей позволяет иметь произвольное количество символов в какой-либо части команды и не записывать нулевые части команд. Применение в качестве символов не только цифр, но и букв, позволяет использовать, кроме действительных, символические адреса.

Символические инструкции, в свою очередь, отделяются одна от другой разделителями.

Совокупность символических инструкций представляет собой исходную программу.

Символическая инструкция по сравнению с обычной машинной командой может дополнительно включать в себя еще два элемента: метку и комментарий.

В полном виде символическая инструкция записывается так:

< МЕТКА > : < ЗНАК > < КОП > , < ИА > , < А1 > , < А2 > < Комментарий > .

Метка при использовании символических адресов играет ту же роль, что и адрес команды при программировании в действительных адресах.

В качестве метки используется только идентификатор.

Идентификатор – последовательность символов, начинающаяся с буквы и не содержащая знаки и разделители.

Пример: A , $A10$, $C\emptyset S$, $SUMMA$

являются идентификаторами ,

$B+1$, $A+B$, $A.$, $2K$ не являются

идентификаторами .

Идентификаторы могут иметь любое количество символов, однако ассемблер различает их по первым шести.

В идентификаторах все пробелы игнорируются.

Метка отделяется от остальных частей инструкции разделителем ":" (двоеточие).

Инструкция может и не иметь метки, в этом случае опускается и двоеточие.

Под знаком понимается обычный алгебраический знак константы или знак кода операции команды (КОП).

КОП символической инструкции записывается 1-6 разрядами КОПа обычной команды – двумя восьмеричными цифрами.

В ассемблере могут использоваться все коды операций ЭВМ "Минск-22" и, кроме того, дополнительно введены коды операций BSS и EQU , назначение которых будет описано ниже.

Для константы под КОП понимаются ее 1-6 разряды.

КОП отделяется от адресных частей разделителем "," (запятая).

Символическая инструкция включает в себя три адресных части - индекс-адрес, первый адрес и второй адрес. Адресные части отделяются друг от друга разделителем ",", (запятая).

В адресных частях символических инструкций допускаются следующие типы адресов:

1. Истинный восьмеричный адрес.

2. Идентификатор.

3. Символ "↑".

4. Алгебраическая сумма предыдущих типов адресов с любым количеством слагаемых, расположенных в любом порядке.

Идентификаторы, используемые в адресных частях, должны быть описаны, т.е. поставлены где-либо в виде метки.

Символ ↑ используется в качестве собственного адреса команды, его содержащей.

Комментарий может содержать все основные символы, кроме "." и "/" (косая черта).

Комментарии отделяются от адресных частей разделителем ",", (запятая).

После комментария ставится признак конца инструкции - разделитель "." (точка). Точка является разделителем между инструкциями .

Замечания.

1. Если истинный адрес и код операции начинаются с нулей, то эти нули писать необязательно .

2. Если код операции положителен, то знак "+" писать не-обязательно.

3. Нулевые части инструкции не пишутся, ставятся только их разделители.

4. Если одна или несколько частей инструкции справа нулевые, то их можно опустить, и точку поставить после самой правой ненулевой части.

Например, команда +0700 0070 0000

в соответствии с замечаниями I-4 может быть записана так:
7.,, 70.

5. Символическим адресам может быть присвоен любой адрес первого и второго МОЗУ, в зависимости от его описания.

Для действительного адреса номер куба указывается либо в разрядах индекс-адреса (т.е. добавляется слагаемое 20, 40 или 60 в индекс-адресе), либо непосредственно в адресных частях инструкции пишется тринадцатиразрядный адрес.

6. В качестве действительного индекс-адреса инструкции используются его младшие шесть разрядов.

Дополнительные команды

I. Одноадресная команда с кодом операции BSS используется для резервации памяти под массивы и для фиксации порядка расположения величин в памяти. Эта команда должна быть обязательно помеченной.

Пример. Последовательность символических команд

W : BSS , 50.

W1 : BSS , 100 означают, что для массива W нужно зарезервировать 50 ячеек, и для массива W1 - 100 ячеек вслед за массивом W .

Символическая команда W I : BSS , W1 - Z .
означает, что размерность массива I зависит от расположения массива Z . Метка команды BSS может быть и фиктивной, т.е. не обозначать какого-либо массива (например, при использо-

вании команды *BSS* для расположения рабочей программы, начиная с заданного адреса).

В адресной части *BSS* допускается любой вышесказанный адрес. Однако все идентификаторы, участвующие в адресе этой команды, должны быть описаны раньше.

2. Одноадресная команда с кодом операции *EQU* применяется для описания некоторых идентификаторов, используемых в программе. В ее адресной части допускается любой вышесказанный тип адреса.

Пример: команда " *RI: EQU , 40* ". означает, что идентификатор *RI* , в тех адресах, куда он входит, при трансляции будет заменен его действительным значением, равным 0040.

При помощи *EQU* можно также указать эквивалентность идентификаторов.

R1: EQU, R2. (*RI* эквивалентен с *R2*)

R IO: EQU , W+10 означает, что идентификатор *IO* эквивалентен десятому элементу массива *W* .

Принцип работы ассемблера

Ввод исходной программы

Ввод перфокарт с исходной программой производится в режиме "копии". Пробитый материал на карте (включая признак конца информации на карте - "/" - косая черта) проверяется на правильность пробивки (число пробивок в колонке должно быть нечетным) и перекодируется.

В случае четного числа пробивок происходит останов с указанием номера карты и неправильно пробитой колонки. Информация, пробитая на карте за символом "/", игнорируется.

Ввод производится массивом по 50 карт до тех пор, пока не встретится признак конца ввода – карта с пробитыми в первых трех колонках символами "/".

Если исходная программа не размещается во втором МОЗУ, то происходит останов.

Трансляция

Трансляция производится в два просмотра исходной программы. При первом просмотре составляется таблица распределения памяти. (ТРП). Память на команды выделяется в том порядке, в котором они стоят в исходной программе.

Под каждую обычную команду или константу выделяется одна ячейка. Под команду с кодом *EQU* память не выделяется. Под команду с кодом *BSS* выделяется массив памяти с размерностью, равной адресу этой команды.

Программа "Ассемблер" построена так, что при первом просмотре ТРП может затирать часть блоков (блок ввода, блок печати исходной программы). Это требует повторного ввода "Ассемблера" при трансляции нескольких программ подряд, но зато увеличивает допустимое количество идентификаторов до 1140_8 .

Если количество идентификаторов меньше 1000_8 , затирания не происходит.

В случае, если число описанных идентификаторов больше 1100 , происходит останов в соответствующим указателем, так как иначе ТРП затрет основные блоки транслятора.

Память распределяется, начиная с ячейки 00100 . Но рабочую программу, используя команды *BSS*, можно поместить в любое место первого или второго МОЗУ.

Например: чтобы поместить рабочую программу с *A*, нужно перед исходной программой поставить команду

< МЕТКА > : BSS, A - < МЕТКА > ,

где 00100 < A ≤ I770.

При втором просмотре программе присваиваются истинные адреса, причем неописанным идентификаторам выделяются ячейки в конце программы, в том порядке, в котором они встречаются в исходной программе. Эти данные включаются в ТРП.

Команды, полученные транслятором, накапливаются , начиная с 100 ячейки. Ячейки, зарезервированные с помощью BSS внутри программы, заполняются нулями. В случае, если длина транслированной программы больше 4400 слов, происходит останов с соответствующим им указателем.

После окончания трансляции происходит останов. Рабочая программа находится с 100 ячейки, независимо от ее места в МОЗУ.

После пуска рабочая программа переносится на заданное место, и управление передается на первую команду.

Выдачи ассемблера

1. Исходная программа распечатывается на АЦПУ. Содержимое каждой перфкарты печатается в одной строке. Печатается также восьмеричный номер карты.

2. Выдача "листинг" на АЦПУ. Выдается рабочая программа. Справа от нее, против каждой команды, печатается соответствующая символическая инструкция исходной программы.

Если в какой-либо инструкции имеется неописанный идентификатор, то слева от соответствующей команды рабочей программы печатается символ "ж" (звездочка) - признак неописанного идентификатора.

Если неописанных идентификаторов в инструкции несколько, то печатается несколько признаков, (но не больше шести).

3. Рабочая программа печатается на ТБПМ группами по IO_8 команд.

Перед каждой группой печатается адрес первой команды.

4. После программы печатается ТРП, по две строки на каждый идентификатор.

В первой строке печатается идентификатор в кодах АЦПУ.

По второму адресу второй строки печатается адрес, соответствующий этому идентификатору.

Идентификаторы, описанные несколько раз, будут столько же раз встречаться в ТРП, однако в программе будет использовано только первое описание идентификатора.

5. Перфорируется рабочая программа с кодом адреса и с контрольной суммой на перфоратор № I.

6. Рабочая программа записывается с контролем на указанную программистом магнитную ленту. Номер шкафа, номер ЛПМ и начальный адрес программы на МЛ должны быть заданы в соответствующих разрядах клавишного набора до трансляции.

Наличие той или иной выдачи определяется одним из ключей (см. таблицу ключей). На ТБПМ независимо от ключей, всегда выдаются три строки, в которых по второму адресу указываются: начало программы, длина ее и адрес первой свободной ячейки МОЗУ после программы.

Если рабочая программа выдается на ТБПМ, то она печатается после этих трех строк.

Подготовка исходной программы

Исходная программа может быть записана на любых бланках, удобных для пользования. (Например, на алгольных или бланках для автокода).

Символы на перфокартах пробиваются по одному в колонке в соответствии с ГОСТ 10859-64 "коды алфавитно-цифровые для перфокарт и перфолент". Количество символов - 64.

Число символов, пробитых на карте, может быть произвольным, однако в конце информации на карте обязательно пробивать символ " / " (косая черта) - признак конца информации.

Пробивка перфокарт производится на устройстве подготовки данных на картах ЭВМ "Минск-22". (ЦПДК).

Можно пробивать перфокарты и на любом другом устройстве, но так, чтобы символы соответствовали данному ГОСТу.

Обычно для обозначения собственного адреса команды применяется символ " ↑ ".

Поскольку этот символ на УПДК "Минск-22" должен набираться из нескольких символов, то при пробивке исходной программы, для упрощения, он заменяется символом "Ю" (основание десятичной системы, во внутреннем коде "Минск-22" - 20₈), которому на телетайпе УПДК соответствует клавиша "?". Это следует учитывать при пробивке перфокарт на других устройствах. При распечатке этот символ печатается как " ↑ ".

Хотя нет никакого ограничения на количество команд на перфокарте, для удобства внесения исправлений следует пробивать на перфокарте целое количество команд, притом небольшое.

Форматная карта перед массивом перфокарт не нужна.

В конце массива обязательно должен быть признак конца массива - карта с пробитыми в первых трех колонках символами " / " .

Таблица ключей

<u>Ключ</u>	<u>Н а з н а ч е н и е</u>
1	Распечатка исходной программы на АЦПУ
2	Трансляция
4	Печать рабочей программы на БПМ
10	Перфорация рабочей программы на перфоратор № 1
20	Печать ТРП на БПМ
40	Запись рабочей программы на МЛ.
100	Выдача листинга программы на АЦПУ.

Таблица остановов

СЧАК	Содержание, см Содержимое, PI	Причины останова	Замечания
664I	Номер колонки Номер карты	Неправильная пробивка	Пробить новую карту. Можно продолжить ввод, нажав пуск.
7006	IIII IIII III7 IIII IIII III7	Исходная прог- рамма не раз- мещается во вто- рой куб	
703I	-7777 7777 7777 -7777 7777 7777	Конец трансля- ции	Пуск. Управление пе- редается на начало программы.
7053	2222 2222 222I 2222 2222 222I	Больше II00 идентификато- ров	
7064	3333 3333 3220 3333 3333 3220	Длина трансля- рованной прог- раммы больше 4400 ₈	Через пуск выдается транслированная часть программы
7420	I7I7 I7I7 I7I7 I7I7 I7I7 I7I7	Неправильная запись на МЛ	Пуск. Повторная запись
7430	I7I7 I7I7 I7I7 I7I7 I7I7 I7I7	Неверное конт- рольное чтение	Пуск. Повторное конт- рольное чтение.

ИНСТРУКЦИЯ ДЛЯ ОПЕРАТОРОВ

Трансляция

1. а) ввод ассемблера с перфоленты - в цикле контрольная сумма - -7777 7777 7777.

б) ввод ассемблера с перфокарт - в автомате после ввода останов. СЧАК 0004.

На P2 контрольная сумма
-7777 7777 7777.

2. Поставить перфокарты с исходной программой и признаком конца.

3. Включите нужные ключи и соответствующие устройства (ТБПМ включается всегда или блокируется).

4. Если нужно записать рабочую программу на МЛ, то набрать на клавишном наборе номер шкафа, номер ЛПМ и начальный адрес программы на МЛ.

5. Пуск с 7000.

6. Останов 703I на СМ и РI

-7777 7777 7777.

Конец трансляции.

7. При других остановах смотреть таблицу остановов.

8. Для передачи управления на начало программы нажать "Пуск". (Если СЧАК сбрасывался, то передать управление в 703I).

9. Следует иметь в виду, что при большом количестве идентификаторов может быть заперчено начало "Ассемблера". (Блок ввода). Поэтому перед повторной трансляцией следует ввести "Ассемблер" заново. "Ассемблер" может быть заперчен также при большом количестве неописанных идентификаторов.

Перфорация ассемблера

1. Ввод ассемблера. Контрольная сумма -7777 7777 7777.
2. Включить перфоратор № I.
3. Передать управление в 722I.
4. Останов 703I.
Конец перфорации.
5. Ввести полученную перфоленту. Контрольная сумма должна быть равна -7777 7777 7777.

Автор благодарен Е.Б.Озерову за многочисленные советы и предложения и за участие в различных этапах работы; В.И.Кочкину и Е.А.Логиновой за внимание к данной работе, а также В.А.Загинайко за полезные советы при работе над программой.

ЛИТЕРАТУРА

1. Загинайко В.А., Силин И.Н.
Автокод "Ассемблер", Дубна, 1968, ОИЯИ БI-II-45I4.
2. Загинайко В.А.
Инвариантное программирование на машины М-20, Минск-22, БЭСМ-6. Препринт ОИЯИ PII-3993, Дубна, 1963.
3. Л.С.Нефедьева, Б.Жаргал
Интерпретирующая система МИС-I.
Препринт ОИЯИ II-3960, Дубна, 1968.

Рукопись поступила в издательский отдел
II августа 1969 года.