В-407

сообщения
объединенного
института
ядерных
исследований
дубна

5603/2-79

7/1-80

P11 - 12654

И.К.Взоров

СОВМЕСТНОЕ ИСПОЛЬЗОВАНИЕ НА ЭВМ ЕС-1010 ЯЗЫКОВ ФОРТРАН И АССЕМБЛЕР В СЛУЧАЕ COMMON- БЛОКОВ

1979

P11 - 12654

И.К.Взоров

СОВМЕСТНОЕ ИСПОЛЬЗОВАНИЕ НА ЭВМ ЕС 1010 ЯЗЫКОВ ФОРТРАН И АССЕМБЛЕР В СЛУЧАЕ **COMMON**- БЛОКОВ

Взоров И.К.

P11 - 12654

Совместное использование на ЭВМ ЕС-1010 языков ФОРТРАН и АССЕМБЛЕР в случае COMMON - блоков

Описывается метод доступа к COMMON -блокам фортранных программ из модулей, написанных на Ассемблере. Предлагается программный способ создания COMMON-блоков в ассемблерных программах.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1979

Vsorov I.K.

P11 - 12654

Joint Usage of FORTRAN and ASSEMBLER Languages on the ES-1010 Computer for the Case of COMMON Blocks

A method of access of FORTRAN programs consisting of modules written in ASSEMBLER to COMMON -blocks is described. A program way of creating COMMON -blocks in ASSEMBLER programs is proposed.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1979

© 1979 Объединенный институт ядерных исследований Дубна

Операционная система малой ЭВМ ЕС-1010 допускает возможность составления одной объектной программы из модулей, написанных на различных входных языках. В работах[1,2] был рассмотрен случай, когда такими входными языками являются ФОРТРАН-IV и компоновочный язык АССЕМБЛЕР, а передача информации между "разноязычными" модулями осуществляется посредством формальных и фактических параметров подпрограмм или подпрограмм-функций. Имеется, однако, другая возможность обмена информацией между отдельными программными единицами - с помощью так называемых общих блоков памяти. На ФОРТРАНЕ эти блоки организуются операторами COMMON/NAME/ℓ, где NAME-имя общего блока (оно может отсутствовать - такой блок называется непомеченным), а ℓ - список идентификаторов переменных,массивов, элементов массивов или их описателей[2]. В компоновочном языке АССЕМБЛЕР[3] у ЭВМ ЕС-1010, в отличие от ряда других машин (например, БЭСМ-6 - см.[4]), нет операторов, эквивалентных фортранному COMMON. Тем не менее,и в этом случае имеет смысл общие блоки памяти типа COMMON сделать доступными для тех частей общих программ, которые написаны на АССЕМБЛЕРЕ, а не на ФОРТРАНЕ. В настоящем сообщении описывается способ обращения к COMMON-блокам из ассемблерных подпрограмм, когда главная программа написана на ФОРТРАНЕ, и способ заведения COMMON-блоков в главной программе, написанной на АССЕМБЛЕРЕ.

КОРОТКО О РАБОТЕ ТРАНСЛЯТОРОВ И РЕДАКТОРА СВЯЗЕЙ

Все трансляторы, используемые в настоящее время на ЕС-1010, генерируют "полуготовые" объектные модули со стандартным форматом RB(Relocatable Binary), из которых редактор связей собирает готовую программу. Физически объектные модули состоят из блоков длиной 120 байтов. При записи на минидиск в один сектор помещаются два таких блока. Один или несколько

модулей образуют двоичный файл (файл RB) . С точки зрения содержащейся в нем информации файл RB состоит из заголовка и программных данных. Заголовок содержит сведения о файле (его имя, количество модулей в нем, количество блоков и др.) .

Когда файл RB помещается на минидиске в зоне GI-GO (в нее он может быть записан транслятором, скопирован из других зон или введен с внешнего носителя данных), заголовок заносится в нулевой сектор этой зоны.

Программные данные образуются из так называемых двоичных записей или статей, генерируемых трансляторами при трансляции операторов и инструкций исходного языка. Так, например, фортранному оператору описания общих блоков памяти COMMON NAME ℓ соответствуют две последовательные записи - $\&17B_1 \ldots B_6$ и $\&19\ell_1\ell_2$, где $B_1 \ldots B_6$ - 6 байтов, содержащих имя общего блока в коде EBCDIC (в случаях непомеченного блока $B_1 \ldots B_6$ - символы пробела), а $\ell_1\ell_2$ - 2 байта, в которых указана общая длина блока в байтах [5]. Транслятор с ФОРТРАНА-IV при появлении в главной программе каждого оператора COMMON генерирует пару таких записей и помещает их в объектный модуль вслед за общим сегментом данных (CDS) программы. Этот CDS (вернее, его RB-образ) определенного формата транслятор с ФОРТРАНА-IV создает в начале трансляции главной программы, независимо от дальнейшего ее содержания. При трансляции же подпрограмм типа SUBROUTINE или FUNCTION CDS не создается, и записи COMMON-операторов помещаются в начале соответствующего RB-модуля. Совокупность подобных записей может образовывать и отдельный модуль - при трансляции подпрограммы типа BLOCK DATA.

Редактор связей (LINKD, LINKE [5]) при сборке готовой программы, обнаружив записи вида &17... и &19..., резервирует указываемую во второй из них память для каждого общего блока вслед за ранее созданным CDS (таким образом как бы увеличивая размер последнего). Начальные адреса общих блоков по мере их заведения заносятся редактором связей в CDS в так называемый накопитель (POOL). Запись &137A в RB-формате CDS указывает редактору связей место и адрес (& 7A), по которому начинается накопитель. Таким образом, начальные адреса COMMON--блоков будут располагаться в CDS готовой программы в ячейках с относительными адресами &7A, &7C и т.д. Обращения к элементам общих блоков транслятором интерпретируются как инструкции с косвенным глобальным индексированным способом адресации (IGX).

ОБРАЩЕНИЕ К COMMON-БЛОКАМ ИЗ ПОДПРОГРАММ НА АССЕМБЛЕРЕ

Из сказанного следует, что если главная программа, написанная на ФОРТРАНЕ, имеет вид

```
COMMON /NAME1/ℓ1
COMMON/NAME2/ℓ2
.............
COMMON/NAMEN/ℓN.
Другие операторы ФОРТРАНА
.................
END
```

то подпрограмма на АССЕМБЛЕРЕ, для того, чтобы из нее можно было обратиться к общим блокам памяти NAME1,..., NAMEN, должна иметь следующую структуру:

```
CDSUBR CDS     DUM
       RES,1   &7A
ANAME1 RES     1
ANAME2 RES     1
.....  ...     .
ANAMEN RES     1
       FIN
LDSUBR LDS
       RES     2
.      .       .
. ДИРЕКТИВЫ     .
.      .       .
       FIN
SUBR   LPS     LDSUBR
.      .       .
. ИНСТРУКЦИИ И ДИРЕКТИВЫ.
.      .       .
       FIN
       END
```

После работы редактора связей в CDS готовой программы в ячейках ANAME1,..., ANAMEN будут находиться начальные адреса соответствующих общих блоков NAME1,..., NAMEN. В этом случае доступ к любому элементу списка I-го общего блока памяти (I=1,...,N) в зависимости от типа элемента может производиться следующим образом:

ТИП ЭЛЕМЕНТА	МЕТОД ПЕРЕДАЧИ ЕГО ЗНАЧЕНИЯ	СПОСОБ ОБРАЩЕНИЯ К ЭЛЕМЕНТУ	
		ЧТЕНИЕ	ЗАПИСЬ
INTEGER	РЕГИСТР A	LDA @#ANAMEI,X	STA @#ANAMEI,X
LOGICAL	РЕГИСТР A	LDA @#ANAMEI,X	STA @#ANAMEI,X
REAL	РЕГИСТРЫ E И A	DLD @#ANAMEI,X	DST @#ANAMEI,X
DOUBLE PRECISION	ЯЧЕЙКИ #T0, #T1, #T2	LEA @#ANAMEI,X CLS F:LDDP	LEA @#ANAMEI,X CLS F:STDP
COMPLEX	ЯЧЕЙКИ #T0, #T1, #T2, #T3	LEA @#ANAMEI,X CLS F:LDCX	LEA @#ANAMEI,X CLS F:STCX

В регистр X перед выполнением любой из этих инструкций должно быть занесено число, равное количеству байтов, занимаемых всеми предшествующими элементами данного блока.

Если подпрограмма, помимо общих блоков памяти, обменивается информацией с фортранными модулями еще и с помощью параметров, ее программные секции (LDS и LPS) должны иметь вид, приводимый в [1,2].

ЗАВЕДЕНИЕ COMMON-БЛОКОВ В ПРОГРАММАХ, НАПИСАННЫХ НА АССЕМБЛЕРЕ

Из приведенного выше краткого описания совместной работы транслятора с ФОРТРАНА и редактора связей можно заключить, что для создания в ассемблерных программах общих блоков памяти аналогично тому, как это делается в случае COMMON-блоков в фортранных программах, необходимо:

1/ Для каждого заводимого общего блока образовать пару записей - &17 .... с именем этого блока в коде EBCDIC и &19... с его длиной в байтах.

2/ Из пар таких записей образовать отдельный объектный модуль (по аналогии со случаем трансляции BLOCK DATA).

3/ Этот новый модуль поместить вслед за RB-модулем главной программы.

4/ В CDS ассемблерной программы вставить запись &137A, указывающую редактору связей место, куда должны помещаться начальные адреса общих блоков, и его относительный адрес.

Иными словами, для заведения COMMON-блоков в ассемблерных программах следует, помимо обычной работы, производимой транслятором ASS2 , проделать еще и часть работы, совершаемой транслятором FORTD.

Для выполнения этой дополнительной работы была создана отдельная программа, носящая в соответствии с ее назначением название "COMMON". Программа вносит все необходимые коррективы в находящийся в зоне GI–GO файл RB, оттранслированной с АССЕМБЛЕРА главной программы до того, как будет вызван редактор связей. Работа производится в режиме диалога с пользователем. В начале работы после вызова % C /COMMON и соответствующего сообщения монитора, программа считывает из нулевого сектора зоны GI–GO заголовок файла, затем первый блок первого сектора, в котором находится начало RB-формата CDS программы. После внесения в него записи &137A этот блок снова записывается в зону GI–GO на свое прежнее место. После этого на устройстве, закрепленном за меткой M:OC(операторская пишущая машинка или заменяющий ее дисплей), печатается сообщение COMMON/, продолжая которое, пользователь должен напечатать имя заводимого общего блока (не более 6 символов), косую черту (/), полную длину блока в 16-разрядных словах (десятичное число), закончив сообщение нажатием знака возврата каретки (RC). В случае непомеченного общего блока имя не печатается. Если при ответе допущена ошибка, его следует аннулировать, нажав перед RC знак стирания (.).

На основании полученного ответа программа создает записи $\&17 в_1 \ldots в_6$ и $\&19 \ell_1 \ell_2$, переведя имя блока в код EBCDIC , а его длину в двоичное число байтов, помещает их в буфер и вновь печатает сообщение COMMON/. Диалог повторяется до тех пор, пока в ответ на такое сообщение пользователь, ничего не печатая, сразу же не нажмет RC . По окончании диалога содержимое буфера в виде отдельного модуля записывается в зону GI–GO вслед за последним блоком модуля главной программы, соответственно исправляется и перезаписывается заголовок файла, после чего, напечатав %% END COMMON, программа заканчивает свою работу. После этого должен быть вызван редактор связей LINKD или LINKE, и работа продолжается обычным путем.

Программа COMMON (вместе с буфером) занимает 311 ячеек памяти.

Для того, чтобы в CDS главной программы, написанной на АССЕМБЛЕРЕ, можно было в нужном месте внести запись &137A, а также для того, чтобы этот CDS могли использовать и фортранные модули составленной редактором связей готовой программы, он должен иметь следующий вид:

```
CDPROG CDS
       RES    36          * TWB СУПЕРВИЗОРА.
ATIO   DATA   TIO         * АДРЕС ТАБЛИЦЫ ВВОДА/ВЫВОДА.
       RES    24          * РАБОЧАЯ ЗОНА ПОДПРОГРАММ ВВОДА/ВЫВОДА.
N2COMM EQU    <2*N>       * N - КОЛИЧЕСТВО ОБЩИХ БЛОКОВ.
ANAME1 EQU    $           * ЛИБО  EQU   &7A.
ANAME2 EQU    $+2         * ЛИБО  EQU   &7C.
...... ...    ...         ..................
ANAMEN EQU    $+N2COMM-2
NFILE  DATA   8           * РАВНО 4-КРАТНОМУ ЗНАЧЕНИЮ КОЛИЧЕСТВА
                          * ОБСЛУЖИВАЕМЫХ ФАЙЛОВ ТАБЛИЦЫ ВВОДА/ВЫВОДА.
TIO    EQU    $+N2COMM    * ТАБЛИЦА ВВОДА/ВЫВОДА.
       DATA   CBLO
       DATA   108
       DATA   CBSI
       DATA   105
       DATA   &80
CBLO   EQU    $+N2COMM
       DATA,1 0,&A0
       DATA,1 &90,M:LO
       DATA   BUFIO
       DATA   132         * ПРИ ВЫВОДЕ НА 80-КОЛОНОЧНОЕ АЦПУ ИЛИ ДИСПЛЕЯ
                          * VT-340 МОЖНО ЗАМЕНИТЬ НА  DATA  80  .
       RES    3
CBSI   EQU    $+N2COMM
       DATA,1 0,&A0
       DATA,1 &10,M:SI
       DATA   BUFIO
       DATA   80
       RES    2
.      .      .
.  ДИРЕКТИВЫ  .
.      .      .
BUFIO  EQU    $+N2COMM    * БУФЕР ВВОДА/ВЫВОДА.
       RES,1  132         * ПРИ ВЫВОДЕ НА 80-КОЛОНОЧНОЕ АЦПУ ИЛИ ДИСПЛЕЯ
                          * VT-340 МОЖНО ЗАМЕНИТЬ НА  RES,1 80  .
       FIN
```

Та часть CDS, которая простирается до пометки "ДИРЕКТИВЫ", является обязательной, так как по своей структуре идентична с CDS, генерируемым транслятором с ФОРТРАНА- IV.

Заметим, что ячейки под начальные адреса общих блоков NAME1,...,NAMEN специально не резервируются, поскольку, как уже говорилось, начальные адреса заводимых COMMON-блоков создаются редактором связей лишь при сборке готовой программы. Для возможности же обращения к общим блокам (метод обращения был приведен в таблице предыдущего раздела) их начальные адреса в исходной программе следует определять через директивы, не производящие резервирования памяти, например, EQU &7A, EQU &7C и т.д. После встраивания накопителя с начальными адресами в оттранслированный CDS истинные положения следующих за ним ячеек окажутся смещенными относительно их первоначальных адресов. Поэтому для коррекции такого смещения первоначальные адреса с самого начала увеличиваются на удвоенное* число заводимых общих блоков N2COMM с помощью директив типа EQU $+N2COMM. Это следует иметь в виду и при заполнении части "ДИРЕКТИВЫ", отведенной пользователю.

В заключение отметим, что при предлагаемом способе создания COMMON-блоков в ассемблерных программах удается избежать вмешательства в саму систему программирования ЕС-1010 (например, посредством введения в язык АССЕМБЛЕР новых директив, эквивалентных фортранному оператору COMMON, или за счет переделки редактора связей). В то же время таким путем довольно просто осуществляется возможность обмена информацией через общие блоки памяти между ассемблерными и фортранными модулями общей программы. Тем самым значительно расширяется область совместного использования этих двух языков при программировании на ЭВМ ЕС-1010.

ЛИТЕРАТУРА

1. Вечеи Т. Совместное использование языка FORTRAN и компоновочного языка в программировании на ЭВМ ЕС-1010. Бюллетень по системам математического обеспечения завода Видеотон, 75/1, стр. 38.
2. FORTRAN IV. Руководство пользователя. 2.03.005.11. 02-SW. Будапешт, 1976.
3. Компоновочный язык АССЕМБЛЕР. Руководство пользователя. 2.03.04.10.02 - SW. Будапешт, 1974.

---

*Так как у ЭВМ ЕС-1010 адресация побайтная.

4. Мазный Г.Л. Программирование на БЭСМ-6 в системе "Дубна". "Наука", М., 1978, стр. 190.
5. Редакторы связей. Руководство пользователя.2.02.09.12.02--SW. Будапешт, 1974.

Рукопись поступила в издательский отдел
12 июля 1979 года.