

СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

4098/83

8/8 83

P11-83-273

А.А.Хошенко

ОБ ОДНОМ РАСШИРЕНИИ КЛАССА
ЭЛЕМЕНТАРНЫХ ОБЪЕКТОВ ДАННЫХ
В ЯЗЫКАХ ТИПА CDL

1983

Языком типа CDL будем считать язык, который (в отличие, например, от Фортрана) не предоставляет программисту никаких встроенных операций над объектами данных, а требует, чтобы все эти операции (например, сложение, умножение, пересылка, сдвиг и т.д.) были заданы в виде макросов.

К языкам типа CDL относятся языки CDL-1 /1/, CDL-1,5 /2/, CDL-2 /3/.

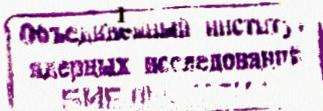
Особенностью (и существенным недостатком) известных к настоящему времени языков типа CDL является то, что в этих языках единственным элементарным типом данных является слово (ассоциируемое обычно с машинным словом целевой ЭВМ, т.е. ЭВМ, на которой предполагается выполнение написанной на CDL программы). Такие языки не позволяют непосредственно задавать в виде макросов операции над такими элементарными объектами данных, как байт, двойное слово, строка байтов, регистр и т.д. По этой причине при необходимости иметь на языке типа CDL эффективную программу, учитывающую особенности машинного языка целевой ЭВМ, приходится прибегать к так называемым "грязным трюкам" /4/.

Предлагаемый в данной работе метод расширения языков типа CDL устраняет перечисленные недостатки этой группы языков. Этот метод позволяет:

- 1) задавать операции над программно-доступными регистрами ЭВМ;
- 2) учитывать наличие в ЭВМ нескольких типов элементарных данных, (например, байт, бит, вектор и т.д.);
- 3) эффективно учитывать способ размещения объекта в памяти, например, странично-адресуемый объект, объект ПЗУ и т.д.;
- 4) задавать начальные значения объектов с учетом возможностей, предоставляемых операционными системами /5/.

Элементарные объекты данных

В машинном языке целевой ЭВМ имеется, как правило, не один тип элементарных объектов данных. В БЭСМ-6, например, такими объектами являются 48-разрядное слово оперативной памяти и 15-разрядные регистры.



Элементарные объекты данных машинного языка ЭВМ БЭСМ-6 WORD (слово) и REGISTER (регистр) могут быть описаны в предлагаемом расширении языка типа CDL следующим образом:

```
STANDARD ELEMENT WORD = 48.  
ELEMENT REGISTER=15.
```

Для машинного языка ЭВМ с байтовой структурой (например, СМ-4) соответствующие элементарные объекты данных BYTE (байт), WORD (слово) и DWORD (двойное слово) могут быть описаны следующим образом:

```
ELEMENT BYTE = 8.  
STANDARD ELEMENT WORD = 2 * BYTE .  
ELEMENT DWORD = 2 * WORD .
```

При этом атрибут STANDARD указывает, какой из элементарных объектов соответствует слову языка типа CDL.

Для каждого элементарного объекта определяем его длину как количество бит, отводимых объекту в памяти ЭВМ.

Размещение элементарных объектов данных в памяти

Описание размещения нестандартных элементарных объектов в расширенном языке типа CDL может быть проиллюстрировано следующими примерами.

Пусть на автокоде (на ассемблере) целевой ЭВМ выделение N байтов памяти для переменной X описывается оператором

```
X.BYTE N.
```

Тогда на расширенном языке типа CDL оператор (макрос)

```
BYTE BYTALLOC(X,Y)=X".BYTE _"Y
```

описывает размещение в памяти объекта типа BYTE , а оператор

```
Y(10):BYTE BYTALLOC
```

сообщает, что Y - элементарный объект (массив из 10 элементов) типа BYTE , и что его размещение в памяти описывается макросом с именем BYTALLOC.

Задание постоянных элементарных объектов (констант)

В языках типа CDL в качестве констант могут быть описаны только объекты, являющиеся словами или группами слов. В предлагаемом расширении языков типа CDL такое ограничение снимается. Проиллюстрируем это следующими примерами.

Пусть на ассемблере целевой ЭВМ задание байтовой константы X, равной C, описывается оператором

```
X.BYTE=C.
```

Тогда на расширенном языке типа CDL макрос

```
BYTE BYTCONST(X,Y)= X".BYTE _"Y.
```

описывает задание байтовой константы, а оператор

```
B100:BYTE BYTCONST=100.
```

присваивает элементарному объекту B100 типа BYTE значение 100.

В реализациях языков типа CDL существует единственный способ внутреннего представления текстовых констант.

Данное расширение позволяет пользователю задавать свой, наиболее подходящий в конкретной ситуации, способ представления таких констант.

Допустим, что на ассемблере целевой ЭВМ текстовая константа с именем X и значением "ABC" описывается оператором

```
X. TEXT _ЗНАВС.
```

Тогда можно задать макрос

```
BYTE TEXTALLOC(X,Y,Z)=X".TEXT _"Y"Н"Z.
```

(X - имя константы, Y - длина (число символов), Z - цепочка символов) и с его помощью описать текстовую константу с именем WP и значением "WHITE _PEN" оператором расширенного языка

```
WP( ):BYTE TEXTALLOC="WHITE _PEN".
```

Идущие подряд скобки означают, что длина константы будет вычислена транслятором и использована как фактическое значение параметра Y макроса TEXTALLOC.

В итоге оператор расширенного языка будет преобразован в оператор ассемблера:

```
WP.TEXT 9НWHITE _PEN.
```

В языках типа CDL невозможно описать целый ряд специфических объектов данных, таких, например, как указатель на странично-размещаемый текстовый буфер файла. Покажем, как это можно сделать на расширенном языке.

```
K=1024.
```

```
FILEBUFPONTER: WORD _POINTALLOC = TXTBUF.
```

```
WORD POINTALLOC(X,Y)=X".DATA _"Y.
```

```
TXTBUF(K): BYTE PAGEALLOC.
```

```
BYTE PAGEALLOC(X,Y)=X".PAGE _"Y.
```

Здесь K - длина буфера TXTBUF в байтах, FILEBUFPONTER - имя указателя этого буфера (тип указателя - WORD), PAGEALLOC - макрос, описывающий странично-размещаемый байтовый буфер с именем X и длиной Y , POINTALLOC - макрос, описывающий объект типа WORD с именем X и значением Y .

Инициализация элементарных объектов

Средства инициализации элементарных объектов в расширенном языке типа CDL могут быть проиллюстрированы на примере задания начального значения байтовой переменной Y, равного символу "A".

```
Y:BYTE BYTALLOC BYTINIT "A".
```

(макрос BYTALLOC был определен выше). Макрос BYTINIT можно определить следующим образом:

```
BYTE BYTINIT(X,Y)="LDB="Y  
"STB" X.
```

Доступ к элементарным объектам

В расширенном языке типа CDL доступ к элементарным объектам для выполнения операций над ними может быть обеспечен известным аппаратом родовых макросов, в которых в качестве родовых параметров могут выступать типы элементарных объектов, и средствами условной трансляции макро-ассемблера целевой ЭЕМ.

Проиллюстрируем указанную возможность расширенного языка на примере операции присвоения.

Пусть на целевой ЭЕМ существуют элементарные объекты типа WORD и REGISTER, и на ассемблере этой ЭЕМ имеются следующие операции присваивания:

```
XTA X  
ATX Y
```

(объекту Y типа WORD присваивается значение объекта того же типа) и

```
MTJ X,Y
```

(объекту Y типа REGISTER присваивается значение объекта X того же типа), и имеются средства условной трансляции следующей формы:

```
IF <условие>  
...  
ELSE  
...
```

Тогда родовой макрос присвоения для указанного набора типов элементарных объектов будет выглядеть так:

```
PROCEDURE MAKE(X:TYPE,Y:TYPE)=  
IF (TYPE=WORD)  
"XTA_" X  
"ATX_" Y  
ELSE  
"MTJ_" X,"Y.
```

Автор выражает благодарность А.А.Корнейчуку за помощь в работе.

Литература

1. Макаренкова А.Д., Назаров Ю.А., Хошенко А.А. "Программирование", 1980, "Наука", М., 1980, № 3.
2. Зен Ы Зон, Ли Рён Хи, Хошенко А.А. ОИЯИ, PII-83-25, Дубна, 1983.
3. Koster C.H.A. Lecture notes in Computer Science, No 47, Springer Verlag, Berlin, 1977.
4. Sandor Barany. AFFIXES, 1982, v. IV, No 1, Netherlands, Nejmegen, Budapest, SZAMKI.
5. Мазный Г.Л. Программирование на БЭСМ-6 в системе "Дубна". "Наука", М., 1978.

Рукопись поступила в издательский отдел
26 апреля 1983 года.

Хошенко А.А.

P11-83-273

Об одном расширении класса элементарных объектов данных
в языках типа CDL

Предлагаемый метод расширения снизу языков типа CDL (CDL-1, CDL-1,5, CDL-2) элементарными объектами данных позволяет задавать операции над программно-доступными регистрами ЭВМ, учитывать наличие в ЭВМ нескольких типов элементарных данных /например, байт, бит, вектор/, эффективно учитывать способ размещения объекта в памяти /например, странично-адресуемый объект/, задавать начальные значения объектов с учетом возможностей операционных систем.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1983

Hoshenko A.A.

P11-83-273

An Extension of Class of Data Elementary Objects
in CDL-Type Languages

The bottom-up extension method of programming languages of CDL-type (CDL-1, CDL-1,5, CDL-2) by data elementary objects permits to define operations over program-accessed computer registers, to use existing in the computer elementary objects of any types, for example, byte, bit, vector, to take into account efficiently the way of arrangement of the object into memory, for example, page-addressed object, to initialize data objects making allowance for operational systems.

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

Communication of the Joint Institute for Nuclear Research. Dubna 1983

Перевод О.С.Виноградовой.