

РАСШИРЕНИЕ ЯЗЫКА ПЛ/1 ПОСРЕДСТВОМ ДИНАМИЧЕСКОГО ВЫЗОВА УТИЛИТ И МАКРОКОМАНД ОБСЛУЖИВАНИЯ КАТАЛОГА И ОГЛАВЛЕНИЯ ТОМА

В последнее время из-за усложнения логической структуры программных систем возникла тенденция реализации большей части их программ на языках высокого уровня. Эти языки имеют развитую структуру (например, обладают конструкциями IF—THEN—ELSE, DO—WHILE, имеют блочную структуру и т. д.), на них алгоритмы легче программируются, они позволяют уменьшить время написания, проверки и отладки программ.

Основными недостатками использования языков высокого уровня являются: отсутствие средств для динамической обработки системных таблиц и данных (например, модификация таблиц, отображающих информацию операторов языка управления заданиями) и возможное замедление выполнения программ. Для ликвидации этих недостатков существуют несколько путей:

а) совместное использование языков высокого и низкого уровня для написания систем. Недостатком этого подхода является необходимость владения разработчиками обоими типами языков;

б) расширение языка Ассемблера макрооператорами, эквивалентными нехватящим конструкциям языка высокого уровня. Примером такого подхода может служить язык СПОК—2, разработанный для ЕС ЭВМ;

в) создание языков высокого уровня предназначенных для системного программирования, таких, как BCPL, PL/360, PL/S, LIS, BLISS, LSD, SUE.

г) расширение распространенных языков высокого уровня общего назначения, (например ПЛ/1, ПАСКАЛЬ и т. д.). Этот подход был применен для написания базовой диалоговой системы автоматизации научных исследований АНИ—81, разработанной в ВЦ АН Армянской ССР. Для этой цели был создан универсальный пакет сервисных процедур ПСП—ПЛ/1.

В данной статье дан метод расширения средств языка ПЛ/1 для использования системных макрокоманд обслуживания каталога и оглавления тома, динамического вызова утилит. Расширение средств языка было достигнуто путем добавления новых разделов в библиотеке SYS1.PL1LIB.

Использование системных макрокоманд обслуживания каталога

и оглавления тома дает следующие возможности: чтение блока из каталога, построение и уничтожение индексной структуры, построение индекса поколений, присвоение и уничтожение псевдонима, соединение и разъединение управляющих томов, каталогизация, повторная каталогизация и раскаталогизация набора данных, чтение блока управления набором данных (DSCB) из оглавления тома, уничтожение и переименование набора данных.

Программы, осуществляющие вышеупомянутые функции, представляют собой совместно отредактированные программы на языках ПЛ/1 и Ассемблера. Программы, написанные на языке ПЛ/1, являются интерфейсом для преобразования заданных пользователем данных к удобному для программ на языке Ассемблера виду. Программы, написанные на языке Ассемблера, выравнивают соответствующие данные, с помощью макрокоманды CAMLST создают список параметров, с помощью соответствующей макрокоманды, описанной в [1], осуществляют требуемое действие и возвращают в программу пользователя код уровня ошибки.

Пример 1

```
SCRATCH: PROC (DSNAME, DATE=IGNORE, VOL=NUMBER,
VOL=LIST, VOL=ERROR=LIST) RETURNS (BIN FIXED (31));
```

```
(*ИНТЕРФЕЙС ДЛЯ ПРОГРАММЫ SCRATCH, УДАЛЯЮЩЕЙ
НАБОРЫ ДАННЫХ*)
```

```
DCL DSNAME CHAR (*) VAR, /*ИМЯ УДАЛЯЕМОГО НАБОРА
ДАННЫХ*/
```

```
DATE=IGNORE BIT(1), /*ПРИЗНАК ИГНОРИРОВАНИЯ СРОКА
ХРАНЕНИЯ*/
```

```
/*ЕСЛИ ПРИЗНАК=1 ТО ИГНОРИРОВАТЬ*/
```

```
VOL=NUMBER BIN FIXED, /*КОЛИЧЕСТВО ТОМОВ, СОДЕР-
ЖАЩИХ УДАЛЯЕМЫЙ НАБОР ДАННЫХ (НЕ БОЛЬШЕ 15)*/
```

```
VOL=ERROR=LIST (VOL=NUMBER) BIN FIXED, (*СПИСОК
КОДОВ УСЛОВИЙ ДЛЯ ТОМОВ*)
```

```
1 VOL=LIST (VOL=NUMBER), /*СПИСОК ТОМОВ (МАССИВ
СТРУКТУР)*/
```

```
2 DEV=TYPE BIT(32) ALIGNED, /*ТИП УСТРОЙСТВА*/
```

```
2 VOL=SER CHAR (6), /*РЕГИСТРАЦИОННЫЙ НОМЕР ТОМА*/
```

```
2 DATA=SET=NUMBER BIN FIXED, /*ПОРЯДКОВЫЙ НОМЕР
НАБОРА НА ТОМЕ. ДЛЯ ДИСКОВ УСТАНОВИТЬ В 0.*/
```

```
ERROR BIN FIXED (31); /* КОД УРОВНЯ ОШИБКИ, ВОЗВРА-
ЩАЕМЫЙ ПРОГРАММЕ ПОЛЬЗОВАТЕЛЯ В КАЧЕСТВЕ ЗНАЧЕНИЯ
ВСТРОЕННОЙ ФУНКЦИИ SCRATCH: /*НИЖЕ СЛЕДУЕТ ОБЪЯВ-
ЛЕНИЯ РАБОЧИХ ПЕРЕМЕННЫХ, КОТОРЫЕ СЛУЖАТ ДЛЯ
ВЫРАВНИВАНИЯ, ХРАНЕНИЯ, СЖАТИЯ ДАННЫХ И УДОБСТВА
ПЕРЕДАЧИ АДРЕСОВ ДАННЫХ ПРОГРАММЕ SCRATCH*/
```

```
DCL WORK=DSNAME CHAR (44),
```

```
WORK=DATE=IGNORE BIT(1) ALIGNED,
```

```

1 WORK-VOL-LIST (VOL-NUMBER),
2 WORK-DEV-TYPE BIT (32) ALIGNED,
2 WORK-VOL-SER CHAR (6),
2 WORK-DATA-SET-NUMBER BIT (16) ALIGNED,
WORK1 DEC FIXED BASED (Q1),
WORK2 DEC FIXED BASED (Q2),
WORK3 DEC FIXED BASED (Q3), /*ПРИСВОЕНИЕ РАБОЧИМ
ПЕРЕМЕННЫМ СООТВЕТСТВУЮЩИХ ЗНАЧЕНИЙ*/
WORK-DSNAME=DSNAME;
WORK-DATE-IGNORE=DATE-IGNORE;
WORK-VOL-LIST=VOL LIST;
Q1=ADDR (WORK-DSNAME);
Q2=ADDR (WORK-DATE-IGNORE);
Q3=ADDR (WORK-VOL-LIST);
VOL-ERROR-LIST=0;
CALL SCRAS (WORK1, WORK2, VOL-NUMBER, WORK3,
VOL-ERROR-LIST, ERROR),
RETURN (ERROR);
END SCRATCH;

```

Пример 2

```

SCRAS START 0
      SAVE (14, 12)
      BALR 2,0
      USING *, 2
      SR 0,0
      LM 3, 8, 0(1)

```

* ПОСЛЕ ВЫПОЛНЕНИЯ ЭТОЙ КОМАНДЫ

* В РЕГИСТРЕ 3 НАХОДИТСЯ АДРЕС ИМЕНИ НАБОРА ДАННЫХ;

* "—" 4	ПРИЗН. ИГНОРИ-
РОВАНИЯ СРОКА ХРАНЕНИЯ;	

* "—" 5	ПАРАМЕТРА КО-
ЛИЧЕСТВО ТОМОВ';	

* "—" 6	СПИСКА ТОМОВ;
* "—" 7	СПИСКА КОДОВ

УСЛ-ИЙ ДЛЯ ТОМОВ;	
* "—" 8	КОДА УРОВНЯ

ОШИБКИ.

```

ST 13, SAVE13
LA 13, SAVEAREA
STM 6, 8, SAVE678
MVC DSNAME(44), 0(3)
MVC NUMBER(2), 0(5)
MVC VOLIST(180), 0(6)
TM 0(4), B'10000000' ПРИЗНАК ИГНОРИРОВАНИЯ=1 ?

```

BO	IGNORE ЕСЛИ ДА, ТО ИДТИ К IGNORE
SCRATCH	NOVRD
B	CONTINUE
IGNORE SCRATCH	OVRD
CONTINUE LM	6, 8, SAVE678
STH	15, 0(8) ЗАПИСЬ В ПАМЯТЬ КОДА УРОВНЯ ОШИБКИ
LH	9, NUMBER ЗАГРУЗКА КОЛИЧЕСТВА ТОМОВ В 9-ЫЙ РЕГИСТР
CONDCODE MVC	1(1,7), 0(6) ЗАПИСЬ В СПИСОК КОДОВ УСЛ-ИЙ ДЛЯ ТОМОВ
LA	7, 2(7) ИЗМЕНЕНИЕ ПЕРЕМЕННЫХ
LA	6, 12(6) ЦИКЛА
BCT	9, CONDCODE
L	13, SAVE13
RETURN	(14, 12)
SAVE678 DS	3F
SAVEAREA DS	1 F
NUMBER DS	H
VOLIST DS	15CL12
OVRD CAMLST	SCRATCH, DSNAME, NUMBER, OVRD
NOVRD CAMLST	SCRATCH, DSNAME, NUMBER
END	

Пример 3

```

OPENJ  START  0
      SAVE   (14, 12)
      ...
      ...
      RDJFCB  MF=(E, ADR)
      ...
ВОССТАНОВЛЕНИЕ DCB
      ...
ИЗМЕНЕНИЕ ПЕРВЫХ 44 БАЙТОВ JFCB
      ...
      OC   JFCB+76(1), YES УСТАНОВИТЬ МАСКУ ДЛЯ ЗА-
                                ПИСИ JFCB ОБРАТНО
      OPEN (DCB), TYPE=J, MF=(E, ADR)
      ...
      ...
      RETURN (14, 12)
DCB    DCB ..., EXIST=LIST
      DS OF
ADR    RDJFCB (DCB), MF=L
      DS OF

```

```

LIST      DC X'07'
          DC AL3 (JFCB)
JFCB      DS CL176
YES       DC B'10000000'
          ...
          ...
          END

```

В примерах 1 и 2 приведены тексты двух программ, предназначенных для расширения средств языка ПЛ/1 средством динамического удаления набора данных.

После того как загрузочный модуль, полученный при совместном редактировании этих двух программ, будет помещен в раздел SCRATCH библиотеки SYS1.PL1LIB, к процедуре функции, удаляющей наборы данных из программы пользователя, можно обращаться следующим образом:

```

ERROR=SCRATCH (DSNAME, DATE=IGNORE, VOL=NUMBER,
               VOL=LIST, VOL=ERROR=LIST).

```

Аргумент DSNAME имеет атрибут CHAR(N), где $N \leq 44$. Размерности массива структур VOL=LIST и массива VOL=ERROR=LIST ограничены числом 15. Атрибуты остальных аргументов и их логический смысл понятны из текста программы, представленной в примере 1. Интерпретация значений кода уровня ошибки и кодов условий для томов приведена в [1, с. 110].

Динамический вызов утилит из программы пользователя дает возможность использования таких основных функций, как:

- а) копирование наборов данных с последовательной, библиотечной, прямой и индексно-последовательной организациями;
- б) копирование записей из набора данных с последовательной организацией;
- в) преобразование последовательной организации набора данных в библиотечную;
- г) изменение символических библиотек;
- д) печатание и перфорация записей из наборов данных с последовательной или библиотечной организациями;
- е) сравнение наборов данных;
- ж) распечатывание системных управляющих данных.

Для выполнения этих функций программа пользователя должна записать соответствующие операторы утилит в набор данных, записи которых являются образами перфокарты, и вызвать программу, написанную на языке Ассемблера, которая с помощью макрокоманды ATTACH или LINK передает управление утилите. При этом утилите передается следующая информация:

- а) информация, указываемая в поле PARM оператора EXEC;
- б) имена, определяющие используемые во время работы утилиты наборы данных. Форматы полей, используемых для передачи информации утилите, приведены в [2].

Так, с помощью динамического вызова утилиты IENMOVE можно копировать и тем самым создавать однотипные наборы данных в процессе работы программы. Однако для осуществления ввода-вывода на логическом уровне с динамически созданными наборами данных требуется иницирование блока управления данными, для чего необходимо еще до начала задания описать эти наборы данных на языке JCL. Обойти эту трудность можно следующим образом:

а) в начале задания с помощью оператора языка JCL описать некоторый фиктивный набор данных;

б) считать в память блок управления файлом задания /JFCB/ описанного набора данных с помощью макрокоманды RDJFCB. Описание этой макрокоманды приведено в [3];

в) модифицировать JFCB и блок управления данными DCB. Форматы этих управляющих блоков приведены в [4];

г) иницировать блок управления данными требуемого набора с помощью макрокоманды OPEN с параметром TYPE=J;

д) осуществление ввода-вывода;

е) выполнение заключительных операций обработки набора данных с помощью макрокоманды CLOSE;

ж) приведение полей блока DCB к исходному виду (так как после использования макрокоманды OPEN с параметром TYPE=J макрокоманда CLOSE неполностью восстанавливает их).

Схема программы для иницирования блока DCB приведена в примере 3.

Описанными средствами следует пользоваться, соблюдая особую осторожность, так как обычные средства контроля в данном случае не действуют.

Ռ. Է. ՀԱՐՈՒԹՅՈՒՆՅԱՆ, Ս. Ս. ՊՈՂՈՍՅԱՆ

PL/I լեզվի հնադավորությունների ընդլայնումը ռաիլիտաների,
դինամիկ կանչերի և տվյալների դեկլարման և կատալոգի
մակրոհրամանների միջոցով

Ա մ փ ո փ ու մ

Հոդվածում տրված է PL/I լեզվի միջոցների ընդլայնման մեթոդը, որը
թույլ է տալիս օգտագործել կատալոգի և հատորի ցանկի սպասարկման
համակարգային մակրոհրամանները, ինչպես նաև ռաիլիտաների դինամիկ
կանչը: Այդ մեթոդը թույլ է տալիս բլոկների ընթերցում կատալոգից, սե-
րունդների ինդեքսի կառուցում, ինչպես նաև տվյալների հավաքածուի վերան-
վանում, հեռացում և այլն:

ЛИТЕРАТУРА

1. Единая система электронных вычислительных машин.
ОПЕРАЦИОННАЯ СИСТЕМА. Дополнительные возможности. Руководство системного программиста. Ц51.804.002 Д35, ч. I.
2. Единая система электронных вычислительных машин. ОПЕРАЦИОННАЯ СИСТЕМА. Утилиты. Введение. Руководство программиста. Ц51.804.002 Д12.
3. Единая система электронных вычислительных машин.
ОПЕРАЦИОННАЯ СИСТЕМА. Дополнительные возможности. Руководство системного программиста. Ц51.804.002 Д35, ч. II.
4. Единая система электронных вычислительных машин. ОПЕРАЦИОННАЯ СИСТЕМА. Управляющие блоки системы. Введение. Руководство системного программиста. Ц51.804.001-01 Д37.