

1212/82

9/III 82



ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

P11-81-751

В.А.Ростовцев

РАБОТА С ВНЕШНИМИ ФАЙЛАМИ
В СИСТЕМЕ **REDUCE-2**

Направлено на Европейскую конференцию
по алгебраическим преобразованиям на ЭВМ
/Марсель, апрель 1982 г./

1981

Система программирования алгебраических преобразований REDUCE -2 /1/ имеет простые средства для хранения информации на внешних файлах. Эти средства представлены командами IN , OUT, SHUT, с помощью которых пользователь может выводить информацию из своей программы на внешние файлы и вводить ее оттуда в оперативную память. По существу, эти команды являются перефразировкой в соответствующем контексте ЛИСП-функций RDS, WRS и CLOSE, то есть просто позволяют пользователю переключать входной и выходной поток системы на различные файлы. При этом информация на этих файлах хранится в символьном виде /в данном случае - в виде исходных выражений языка REDUCE или в виде текстов, сформированных программой пользователя/. Эти средства системы REDUCE-2 весьма полезны для пользователя, однако обладают, по меньшей мере, двумя недостатками. Первый связан с символьной формой хранения информации. Если на внешнем файле хранится программа, предназначенная для повторного исполнения, то при формировании такого файла система вынуждена преобразовывать выражения из внутреннего спискового представления в символьный текст. При многократном вводе такого файла каждый раз исходный символьный текст транслируется вновь во внутреннее списковое представление. Это приводит к излишней трате машинного времени. Кроме того, неэкономно расходуется память на диске. Вторым недостатком определяется последовательной организацией входного потока системы и непрерывностью его обработки. Пользователь может обрабатывать большой входной файл по частям только путем разбиения его на некоторое число более мелких файлов, каждый из которых вводится в память и обрабатывается целиком. Даже если сам алгоритм исходной программы и позволяет выполнить такое разбиение, это создает большие неудобства для пользователя, вынужденного поддерживать утомительную "бухгалтерию" файлов. Часто же логика решаемой задачи вообще не позволяет этого сделать. Невозможно также создавать библиотеки стандартных процедур в классическом смысле, то есть такие библиотеки, из которых каждая конкретная программа пользователя выбирает лишь необходимые ей процедуры. Наконец, такой метод обработки входного потока может привести к серьезным затруднениям при решении задач - если в программе используется много процедур, определенных пользователем, обращения к которым не удастся строго упорядочить, то для хранения этих процедур расходуется значительный объем оперативной памяти.

В настоящем сообщении предлагается дополнить систему REDUCE еще одним средством работы с внешними файлами, в значительной мере свободным от указанных недостатков. Это средство основывается на аппарате виртуальной памяти для функций, имеющемся в интерпретаторе UT LISP 4.1 для машин CYBER 170/CDC 6000-7000^{2,3/}. Доопределение языка Standard LISP^{4/} и реализация подобного аппарата в соответствующих интерпретаторах для других ЭВМ позволит ввести предлагаемое средство в машинно независимую систему REDUCE-2 в качестве элемента ее входного языка.

Таким элементом входного языка системы REDUCE-2 является команда

```
DOUT (<список процедур> [, <имя файла>]);
```

Здесь <список процедур> перечисляет имена предварительно определяемых пользователем процедур, предназначенных для хранения на внешнем файле, а <имя файла> указывает на тот файл, где будут храниться эти процедуры. Заключение последнего параметра в квадратные скобки говорит о том, что он может быть опущен; в этом случае используется файл с именем VIRFN, определенный в интерпретаторе ЛИСПа.

Эта команда реализуется соответствующей добавочной процедурой системы REDUCE-2, аналогично команде OUT, и использует функцию ЛИСПа DISKOUT. Последняя требует кроме списка имен функций еще два числовых параметра^{2/}. В качестве таковых используются значения глобальных переменных !/!/CODEMIN и !/!/CODEMAX. Эти значения устанавливаются при генерации системы REDUCE, но могут и изменяться пользователем при решении его задачи.

При желании пользователь может полностью освободиться от управления виртуализацией процедур. Для этого в системе REDUCE вводится флаг VIRFN, который устанавливается, как обычно, командой ON VIRFN; . Такая команда вызывает автоматический "сброс" всех последующих определений процедур в программе на внешний файл с именем VIRFN. Эта возможность, аналогично другим флагам в REDUCE, реализуется путем введения глобальной переменной !*VIRFN и дополнения соответствующих функций системы /реализующих трансляцию процедур/ обращением к функции ЛИСПа DISKOUT.

Описанные дополнительные элементы входного языка системы REDUCE дают возможность пользователю при исполнении его программы хранить определенные в ней процедуры на внешнем файле; их вызов в оперативную память осуществляется автоматически по обращению. Таким образом, можно освободить оперативную память для размещения собственно обрабатываемых выражений. Во многих случаях будет экономиться также время работы програм-

мы - за счет уменьшения числа сборок мусора. В связи с этим интересен такой пример. В интерпретаторе UT LIST 4.1 стандартный буфер для обмена с внешними файлами занимает 256 слов. Если суммарная длина виртуализуемых функций не превышает этого размера, то фактической записи на диск не происходит, а вся работа выполняется с буфером. Для некоторых задач пользователя, удовлетворяющих этим условиям, применение виртуализации процедур позволило сократить число сборок мусора в 2,5 раза, время выполнения программы - в 2 раза.

Профессор А.Херн^{5/} предлагает применять в некоторых случаях "сохранение операционной среды" системы REDUCE. В сочетании с этим приемом предлагаемое нами использование виртуальной памяти для функций позволяет также организовать библиотеку пользователя или системные библиотеки стандартных процедур. Они представляются особенно удобными для стандартных пакетов типа пакета решения обыкновенных дифференциальных уравнений, пакета решения интегральных уравнений, интегратора и т.п.

ЛИТЕРАТУРА

1. Hearn A.C. REDUCE 2 User's Manual, Univ. of Utah, Salt Lake City, UCP-19, 1973.
2. LISP Reference Manual CDC-6000. The Univ. of Texas at Austin, Computer Center, CCUM-2, 1975.
3. Kadantsev S.G., Rostovtsev V.A. SIGSAM Bull., 1980, 14, p. 14.
4. Griss M.L., Hearn A.C. Software Practice & Experience, 1981, 11, p. 541.
5. Hearn A.C. REDUCE Newsletter, No.7, Univ. of Utah, Salt Lake City, 1979, p. 14.

Рукопись поступила в издательский отдел
27 ноября 1981 года.