

СИНТАКСИЧЕСКИЙ КОНТРОЛЬ ПРОГРАММ  
НА ЯЗЫКЕ ЛИСП ДЛЯ БЭСМ-6

В.С. Лозовский

Язык ЛИСП [1] вот уже около десяти лет с успехом применяется при решении задач, связанных с обработкой преимущественно символьной информации и имеющих достаточно сложную структуру данных. Ряд новых, более гибких и мощных языков в той или иной степени наследуют основные положения ЛИСПа. Интерпретатор и компилятор для ЛИСПа [2] включены в качестве СП в стандартное математическое обеспечение ЭВМ БЭСМ-6 (режим Д-68 [6]). Работы над этой системой в настоящее время успешно продолжаются ([3], [4]).

Характерной чертой языка ЛИСП является широкое использование скобочных выражений, зачастую весьма сложных. Ручной контроль синтаксической правильности при написании и отладке ЛИСП-программ в связи с этим затруднен. В настоящей работе используется синтаксический формат записи, делающий программу более обзорной и облегчающий соблюдение формальной правильности записи, а также приводится программа, реализующая распечатку ЛИСП-текста на АЦПУ. При подключении СП125 (ЛИСП) к БЭСМ-6 в режиме операционной системы ДИСПАК [5] возникли определенные трудности, связанные с имеющимися в системе ограничениями на длину СП. Поэтому пришлось отказаться от использования транслятора ЛИСП как стандартной подпрограммы. Его вызов в оперативную память, ввод и распечатка ЛИСП-текстов осуществляется специальной программой-монитором, описание действующего в настоящее время варианта которого и приводится ниже. В дальнейшем

предполагается сервис, обеспечиваемый монитором, расширить, включив в него работу с архивами и редактирование ЛИСП-текстов.\*)

#### Алгоритм синтаксического контроля монитора

Выбирая наиболее удобный формат представления ЛИСП-текстов, можно руководствоваться либо синтаксическими, либо семантическими соображениями. С точки зрения семантики, удобно как-то выделять обращения к наиболее важным функциям: вынести метки (режим *PROG*) на одну линию и т.п. Ясно, что способ такой записи в большой степени зависит от вкуса программиста, трудно поддается унификации и алгоритмизации. К тому же в ЛИСПе очень часто встречаются выражения, имеющие большую длину и глубину вложенности; при этом физические размеры бланка налагают свои ограничения, часто идущие вразрез с желанием программиста.

Рассмотрим, в качестве примера, функцию *SUBSTRINGP*, принимающую значение *T*, если в списке *Y* присутствуют все элементы списка *X* с сохранением их упорядоченности. Один из вариантов её "семантической" записи представлен на рис. 1.

```

1(SEXP 2SUBSTRINGP (LAMBDA (X Y)
3(COND ((AND (NULL X)(NULL Y)) T)
4((NULL Y) NIL)
5((NULL X) T)
6((EQUAL (CAR X)(CAR Y))
7(SUBSTRINGP (CDR X)(CDR Y)))
8((EQUAL (CAR X)(CADR Y))
9(SUBSTRINGP (CDR X)(CDDR Y)))
10(T (SUBSTRINGP X (CDR Y))))))

```

Рис. 1

\* К сожалению, известные работы, связанные с управлением решением задач и службой архивов, в частности для ЛИПСа [8], не предназначены для работы в системе ДИСПАК [5].

Одним из способов проверки синтаксической правильности написанной программы является приписка скобкам номеров соответственно их глубине по вложенности. При этом глобальный баланс устанавливается по совпадению номеров самой первой и самой последней скобок. Контроль облегчается и внутри. Так, в соответствии с приведенной разметкой скобок видно, что все аргументы функции *COND* занимают четвертый уровень. Этот принцип синтаксического контроля легко кладется в основу формата представления ЛИСП-программ и данных. Достаточно связать номер скобки с её позицией на бланке или машинной выдаче. Предлагаются следующие правила при записи текста на бланке.

1. Первая открывающая скобка ставится в позицию бланка, выбранную в качестве начальной.

2. Очередная открывающая скобка помещается в позицию, определяемую её номером (со смещением вправо).

3. Последовательность открывающих скобок в одной строке может встретиться, лишь когда они следуют подряд. Если вслед за некоторой открывающей скобкой идет текст, то очередная открывающая скобка для сохранения соответствия её номера и позиции должна быть перенесена на очередную строку.

4. Ряд закрывающих скобок с пустыми или непустыми последовательностями символов между ними замыкает соответствующую строку. Принцип соответствия номеров и позиций к ним не применяется.

5. Последовательность символов, стоящих перед очередной открывающей скобкой, помещается на предыдущую строку.

Итак, каждая строка записи непременно начинается открывающей скобкой; строка смещена вправо (по отношению к самой первой) на число позиций, равное разности в глубине вложенности их скобок.

На рис. 2 приведена машинная распечатка текста функции *SUBSTRINGP* в синтаксическом формате и одного из обращений к ней в виде, обеспечиваемом монитором.

В левой из двух колонок чисел, печатаемых с  $17I_{10}$  позиции, указывается восьмеричный номер первой открывающей скобки данной строки. В правой колонке — номер последней закрывающей скобки строки (если она есть; в противном случае — ноль). В отдельные строки выносятся примечания, печатаемые с  $10I_{10}$  позиции и

```

1      01 00 (SEXPR SUBSTRINGP
      02 00 (LAMBDA
      03 03 (X Y)
      03 00 (COND
      04 00 ((AND
      06 06 (NULL X)
      06 04 (NULL Y)) T)

1-1    04 04 ((NULL Y) NIL)
      04 04 ((NULL X) T)
      04 00 ((EQUAL
      06 06 (CAR X)
      06 05 (CAR Y))
      05 00 (SUBSTRINGP
      06 06 (CDR X)
      06 04 (CDR Y)))

2      04 00 ((EQUAL
      06 06 (CAR X)
      06 05 (CADR Y))
      05 00 (SUBSTRINGP
      06 06 (CDR X)
      06 04 (CDDR Y)))
      04 00 (T
      05 00 (SUBSTRINGP X
      06 01 (CDR Y))))))

T 1    01 06 (SUBSTRINGP
      02 00 (QUOTE
      03 02 (A B))
      02 00 (QUOTE
      03 01 (A B C D)))

T 2    01 00 (SUBSTRINGP
      02 00 (QUOTE
      03 02 (A B C))
      02 00 (QUOTE
      03 01 (M N A B C D)))

```

вносимые программистом в текст программы на ЛИСПе по определенным правилам. Каждая строка ЛИСП-текста печатается с номера позиции, определяемого выражением:  $24 + K$ , где  $K$  - номер первой открывающей скобки строки. Монитор позволяет в процессе решения ЛИСП-задачи печатать на АЦПУ метки текущего времени центрального процессора с момента начала счета по данной задаче. Обращение из программы: (TERPRI). Эта возможность реализуется посредством обращения к экстра-коду 063 0004.

Рис. 2

#### Правила записи ЛИСП-текста

Следующий ниже текст правил, естественно, отражает "точку зрения" на ЛИСП-текст монитора и не соответствует строгому синтаксическому описанию языка ЛИСП (если таковой и сделан где-либо).

**ЛИСП ТЕКСТ** - это ЧАСТЬ ПРОГРАММЫ, либо ПРИМЕЧАНИЕ, либо СЛУЖЕБНЫЕ КОМБИНАЦИИ.

**ЧАСТЬ ПРОГРАММЫ** - это последовательность всех допустимых символов АЦПУ [ 2, с. 2-3 ], не содержащая

СЛУЖЕБНЫХ КОМБИНАЦИЙ и начинающаяся круглой открывающей скобкой.

**ПРИМЕЧАНИЕ** - это последовательность всех допустимых символов АЦПУ, кроме круглых открывающих скобок, не содержащая СЛУЖЕБНЫХ КОМБИНАЦИЙ:  $\times$   $10A$   $10B$  и следующая непосредственно за служебной комбинацией  $10 10$ . Примечание, стоящее впереди другого примечания, не может быть пустым. Концом примечания считается символ, стоящий перед круглой открывающей скобкой (началом очередной части программы), комбинацией  $10 10$  или комбинацией:  $\times$ .

**СЛУЖЕБНЫЕ КОМБИНАЦИИ** - это

$\times$  - конец ЛИСП-текста;

$::$  - следующий символ-номер позиции,

$<$  - перевод строки,

$>$  - кратность предыдущего символа,

$10 10$  - (восьмеричный код символов УПП: 020 020) - начало

примечания,

$10A$  - с этого места начать распечатку ЛИСП-текста,

$10B$  - с этого места прекратить распечатку ЛИСП-текста.

Итак, записывая ЛИСП-текст на бланке, рекомендуется придерживаться предлагаемого синтаксического формата.\* Однако это не обязательно. Перфорация ЛИСП-текста производится безотносительно к формату бланка и формату перфокарты: никаких дополнительных пробелов сверх тех, которые требуются синтаксисом, а также наличие любого числа восьмиразрядных пустых позиций внутри карты роли не играет. Никаких специальных символов конца перфокарты нет. Рекомендуется каждую перфокарту начинать примечанием: например, её номером, названием и т.п., с тем, чтобы не обращаться впрямь к тексту на исходных бланках. Если длина текста примечания велика, рекомендуется разбивать его служебными комбинациями  $10 10$ . Число строк примечания равно числу этих комбинаций. Отличие этого способа от употребления комбинации:  $<$  состоит в том, что в последнем случае печать следующего за ней текста начнется с нулевой, а не с  $10 10$  позиции АЦПУ.

\* Это облегчает последующую сверку исходного текста с машинной распечаткой.

Если необходима печать всего ЛИСП-текста, то начать его следует перфокартой со служебной комбинацией  $10A$ , поскольку начальное состояние монитора соответствует выключенной распечатке. Количество и порядок следования в тексте комбинаций  $10A$  и  $10B$  (включающих и выключающих печать) не ограничивается, за исключением запрещения использования их в примечаниях.

#### Описание монитора

Монитор написан на автокоде БЕММ [7] и после трансляции в виде готовой программы помещается в  $I7_8$  зону магнитной ленты. Наглядность автокод-текста освобождает от необходимости приводить блок-схему монитора. Ниже приводятся краткие пояснения.

Монитор начинает работу с ввода карт исправлений (если они есть) и ЛИСП-текста; последний размещается с метки *PR* (адрес 2001). Монитор просматривает ЛИСП-текст посимвольно и перезаписывает его в массивы *PROG* (с адреса 2000) и порциями - в массив *PRSPCE*. В первом случае формируется текст ЛИСП-программы (без примечаний), который затем переписывается в соответствии с требованиями [2] на МБ I-0. Во втором случае формируется текст для печати очередной строки. Это может быть текст примечания, снабженный признаком конца буквенно-цифровой печати или же текст очередной строки ЛИСП-программы; при этом в две ячейки массива *BRACKS* помещаются восьмеричные константы, определяющие номера граничных скобок строки.

#### Использование модификаторов:

M1 - текущий адрес по массиву *PRSPCE*.

M2 - текущий адрес по исходному тексту в массиве *PR*.

M3 - текущий адрес по тексту ЛИСП-программы в массиве *PROG*.

M4 - используется для задания начального адреса текста, печатаемого с использованием информации *PRINT1*.

M7 - для хранения адреса возврата из подпрограммы (кроме *STPRNT*).

M8 - для хранения адреса возврата из *STPRNT*.

#### Перечень основных меток монитора.

*STACK* - ячейка памяти, "окно" в котором всегда находятся очередные шесть символов ЛИСП-текста.

*ADVSYM* - подпрограмма сдвигающая символы в *STACK* на один символ влево (пустая позиция справа, естественно, заполняется очередным символом).

*ACCSTK* - перенос первого символа из *STACK* в массив *PRSPCE*.

*ACCEPT* - дополнительный вход подпрограммы *ACCSTK*: при этом в *PRSPCE* переносится первый символ с думпатора.

*ACCST* - перенос первого символа из *STACK* в "чистый" текст ЛИСП-программы (*PROG*).

*START1* и *STAR1* - входы для установки начальных состояний соответствующих подпрограмм.

*TFIN* - проверка, не является ли первый символ в *STACK* признаком конца ЛИСП-текста: \*x.

*TPRNT* - проверка на наличие комбинации  $10A$ .

*NOPT* - проверка на наличие комбинации  $10B$ .

*TST1* и *TEST1* - проверка на открывающую круглую скобку.

*TST2* - проверка на закрывающую круглую скобку.

*TCOMM* - проверка на комбинацию  $1010$ .

*WRITE* - ЛИСП-программа записывается на МБ I-0. Затем считывается на I лист МСЗУ первая зона ЛИСП-транслятора. По первому информационному слову определяется длина транслятора и вычисляется число зон, которые еще следует считать. После их чтения весь транслятор переписывается в массив с ячейки *BLISP*. На  $I5_8$  модификаторе устанавливается значение базы: *BLISP*, на  $I6$  - адрес информационного слова транслятора: *INFLSP*. Адрес возврата из транслятора: *RETURN* заносится по относительному адресу:  $I$  ( $M I5_8$ ) и передается управление на вход транслятора по относительному адресу:  $IO_8$ . После возврата из транслятора печатается оповещение о конце работы монитора.

Информация, определяющая распределение памяти ЛИСП-транслятора, хранится в ячейке *INFLSP* и может быть изменена с помощью карт исправлений при выходе монитора. В действующем варианте длина участка списочной памяти  $P1$  [2] задается в  $I5$  младших разрядах; следующие  $I5$  разрядов - это длина участка ПНС  $P2$ , затем - длина ассоциативного списка  $P3$ .

На магнитной ленте хранится следующая информация:

- с 0 зоны - автокод-текст [9] ЛИСП-транслятора,

- в I7<sub>8</sub> зоне - рабочая программа монитора *MNTLSP* (с 20 ячеек),

- с 20<sub>8</sub> зоны - рабочая программа ЛИСИ-транслятора.

С помощью небольшой вспомогательной программы вызывается текст ЛИСИ-транслятора<sup>\*</sup>, в него вносятся необходимые изменения, и после трансляции рабочая программа записывается со 2 ячейки 20<sub>8</sub> зоны МЛ. В её нулевой ячейку помещается информация, требуемая при задании первого информационного слова для любой СП [6]; в действующем варианте это:

C 00 00 0006  
2I 20 62I2

Содержимое ячейки I 20 зоны монитором не используется.

Колода при обращении к системе составляется в соответствии с требованиями [5].

ШИФР ХХХХХ

ЛИСТ 0 - I4

ВРЕМ 1500

ВХОД I7

ТРАК 32

АЦПУ 15

ЛЕНТ 30 (НННН - I7)

Карты изменений и дополнений паспортной информации

Е

B 00 000 0016

00 000 0000

K 00 010 0000

00 030 0017

K 00 070 0016

00 000 0000

Карты рабочих исправлений текста *MNTLSP*: например:

\* Текст ЛИСИ интерпретатора/компилятора на автокоде [9] любезно предоставил в наше распоряжение В.М. Офа.

B 00 000 0257  
00 000 0000  
C P3 P2 P1

Изменение распределения памяти

B 00 000 200I  
00 000 0000  
AO

ВВОДНОЕ СЛОВО и признак перекодировки ЛИСИ-текста

ЛИСИ-текст

:x - ◊ ЕКОНЕЦ

СТАНДАРТНЫЙ КОНЕЦ ПАКЕТА ЛИСИ

О правилах записи ЛИСИ-текста говорилось выше. В приложении приведен текст программы монитора *MNTLSP* на автокоде БЭМШ [7].

#### Л и т е р а т у р а

1. MCCARTHY John et al. LISP 1.5 Programmer's Manual, M.I. T. Comp. Cent., Cambridge Massachusetts, 1972.
2. ЛАВРОВ С.С., СИЛАГАДЗЕ Г.С. Входной язык и интерпретатор системы программирования на базе языка ЛИСИ для машины БЭСМ-6. М., 1969, (ВЦ АН СССР).
3. СИЛАГАДЗЕ Г.С. Система программирования с языка ЛИСИ для машины БЭСМ-6. Дис. на соискание учен. степени к.ф.-м.н. М., 1971, (ВЦ АН СССР).
4. ОФА В.М. Развитие системы программирования ЛИСИ-БЭСМ-6. - В сб. "Обработка символической информации". М., 1973, (ВЦ АН СССР).
5. ЗЕЛЬДИНОВА С.А., ЗУЕВ В.И., КОШКИНА Л.В., ОЗОРИН Ю.В., ТЮРИН В.Ф., ШУЛЕПОВ Н.И. Инструкция пользователю по работе с операционной системой ЛИСИ-ПАК для БЭСМ-6, М., 1972, (ИИМ АН СССР).
6. Инструкция по программированию на БЭСМ-6. М., 1971, (Московский завод счетно-аналитических машин).
7. ШТАРКМАН В.С. Математическое обеспечение БЭСМ-6. Автокод БЭМШ. Описание языка. М., 1969, (ИИМ АН СССР).
8. БРЯБРИН В.М., САФОНОВ В.И., ОФА В.М. Система ПУЛЬТ для управления программами на АЛГОЛЕ, ЛИСИ и АВТОКОДЕ. - Сообщения по вычислительной математике, вып. 7, М., 1972 (ВЦ АН СССР).
9. Автокод БЭСМ-6 (Инструкция) М., 1969. (ИИМ и ВТ АН СССР).

Поступила в ред.-изд.отд.  
12 апреля 1973 года

ПРИЛОЖЕНИЕ

ИЛИ БИМН ОТ 27-10-69		СТР 0001	
ПРЕДЛ	МЕТКА	ОПЕРАЦИЯ	ОПЕРАНД
0001	MNTLSP	СТАРТ	'20'
0002		S73	1
0003	MNTR	PRA	3
0004		VNA	HEAD(4)
0005		PEH	PRINT1
0006		VNA	PROR+1(2)
0007		C4	(2)
0008		3П	STACK
0009		ПВ	START2(7)
0010		VNA	PROG(3)
0011		ПВ	STAR1(7)
0012	TFIN	C4	STACK
0013		И	S1
0014		HTM	FIN
0015		PE	TPRNT
0016		ПВ	ACCSTK(7)
0017		ПВ	P2(7)
0018		ПВ	INILSP
0019	TPRNT	C4	STACK
0020		И	S12
0021		HTM	ON
0022		PE	NOPT
0023		ПВ	STPRNT(8)
0024		C4	=U'1'
0025		3П	ONPRNT
0026	ADV2	ПВ	ADVSVH(7)
0027		ПВ	ADVSVH(7)
0028		ПВ	TFIN
0029	NOPT	C4	STACK
0030		И	S12
0031		HTM	OFF
0032		PE	TST1
0033		C4	
0034		3П	ONPRNT
0035		ПВ	ADV2
0036	TST1	C4	STACK
0037		И	S1
0038		HTM	BRCL
0039		PE	PRNBR
0040	INCR	C4	CNTR
0041		CAU	=U'1'
0042		3П	CNTR
0043		C4	FBR
0044		ПВ	TPRBR
0045	INBR	C4	CNTR
0046		3П	BRACKTS
0047		CAU	64-36
0048		CAU	CPRINT
0049		3П	PRINT2+2

КОММЕНТАРИИ

HEADING  
INITIATE ADVSVH  
INIT. ACCST

NOT PRINT ON

NOT PRINT OFF

IF OPENING BRACKET

NOT OP BRACKET  
INCREMENT BRACKET'S NO.

FIX OP BRACKET'S NO.

ИЛИ БИМН ОТ 27-10-69

СТР 0002	
ПРЕДЛ	МЕТКА
0050	C4
0051	3П
0052	C4
0053	3П
	FBR
	=U'1'
	PREVBR
0054	ACC
	ПВ
	ACCSTK(7)
0055	ПВ
	ACCST(7)
0056	ПВ
0057	ПВ
	ADVSVH(7)
	TFIN
0058	TPRBR
0059	C4
0060	PE
0061	C4
0061	ПВ
	ACCSTK(7)
0062	ПВ
	P2(7)
0063	ПВ
0064	ПВ
	STPRNT(8)
	INBR
0065	PRNBR
0066	C4
0067	3П
0068	C4
0069	И
0070	HTM
0071	PE
0072	C4
0073	3П
0074	C4
0075	CAU
0076	3П
0077	ПВ
0078	TCOMM
0079	C4
0080	И
0081	HTM
0082	PE
0082	C4
0082	ПВ
	TAILPR
0083	COMOUT
	ПВ
	STPRNT(8)
0084	ПВ
	ADVSVH(7)
0085	ПВ
0086	TEST1
0087	C4
0088	И
0089	HTM
0090	PE
0091	C4
0091	ПВ
	ADVSVH(7)
	STACK
	S1
	BRCL
	TEND
	FIN
	ACCSTK(7)
0092	ПВ
	P3(7)
0093	ПВ
0094	ПВ
	STPRNT(8)
	INCR
0095	TAILPR
	C4
	FIN

WAS THE PREV SVMB AN OP BRACKET

PREV SYMBOL IS NOT A BRACKET  
IF CLOSING BRACKET

IF COMMENT

84-1  
84-2

PRINT INFORM TAIL

ИЛИ БЕНД ОТ 27-10-69				СТР 0003
ПРЕВН	МЕТКА	ОПЕРАЦИЯ	ОПЕРАНД	КОММЕНТАРИЙ
0096		ПВ	ACCEPT(7)	
0097		ПВ	P2(7)	
0098		ПВ	CONOUT	
0099	РЕНД	СЧ	STACK	IF END
0100		И	S1	
0101		НТМ	FIN	
0102		ПЕ	CONLUP	
0103		ПВ	ACCSTK(7)	
0104		ПВ	P3(7)	
0105		ПВ	INILSP	
0106	CONLUP	ПВ	ACCSTK(7)	ACCUMULATE COMMENT
0107		ПВ	ADVSN(7)	
0108		СЧ	STACK	
0109		И	S12	
0110		НТМ	COMMEN	
0111		ПЕ	TEST1	
0112		СЧ	FIN	
0113		ПВ	ACCEPT(7)	
0114		ПВ	P3(7)	
0115		ПВ	CONOUT	
0116	P2	СЧ	ONPRNT	PRINT INFORM
0117		ПО	(7)	
0118		СЧ	(1)	
0119		ПЕ	PR2	
0120		СЛМА	-1(1)	
0121	PR2	ПЕЧ	PRINT2	
0122		ПВ	(7)	
0123	P3	СЧ	ONPRNT	PRINT COMMENT
0124		ПО	(7)	
0125		ПЕЧ	PRINT3	
0126		ПВ	(7)	
0127	STPRNT	УМА	PRSPCE(1)	STARTING PRINT CONDITIONS
0128		ПВ	START1(7)	
0129		СЧ	УМ-1	
0130		ЗП	PREVDR	
0131		ЗП	FBR	
0132		СЧ		
0133		ЗП	BRACKTS-1	
0134		ПВ	(8)	
0135	ADVSN	СЧ	STACK	
0136		СВА	64-8	
0137		ЗП	STACK	
0138		СЧ	(2)	
0139		СВ	SM2	
0140		И	S6	
0141		ИЛИ	STACK	

ИЛИ БЕНД ОТ 27-10-69				СТР 0004
ПРЕВН	МЕТКА	ОПЕРАЦИЯ	ОПЕРАНД	КОММЕНТАРИЙ
0142		ЗП	STACK	
0143		СЧ	SM2	
0144		ВЧПА	64-8	
0145		ЗП	SM2	
0146		НТМ	MIN	
0147		ПЕ	(7)	
0148	START2	СЛМА	1(2)	
0149		СЧ	ZSM2	
0150		ЗП	SM2	
0151		ПВ	(7)	
0152	ACCSTK	СЧ	STACK	
0153	ACCEPT	И	S1	
0154		СВ	SM	
0155		ИЛИ	(1)	
0156		ЗП	(1)	
0157		СЧ	SM	
0158		СЛПА	64-8	
0159		ЗП	SM	
0160		НТМ	MAX	
0161		ПЕ	(7)	
0162		СЛМА	1(1)	
0163	START1	СЧ		
0164		ЗП	(1)	
0165		СЧ	ZSM	
0166		ЗП	SM	
0167		ПВ	(7)	
0168	ACCST	СЧ	STACK	
0169		И	S1	
0170		СВ	SM+1	
0171		ИЛИ	(3)	
0172		ЗП	(3)	
0173		СЧ	SM+1	
0174		СЛПА	64-8	
0175		ЗП	SM+1	
0176		НТМ	MAX	
0177		ПЕ	(7)	
0178		СЛМА	1(3)	
0179	STAR1	СЧ		
0180		ЗП	(3)	
0181		СЧ	ZSM	
0182		ЗП	SM+1	
0183		ПВ	(7)	
0184	INILSP	УМА	PRO(1)	
0185		СЧМ		
0186		СВА	64-10	
0187		ВЧОВ		
0188		УМ	2	
0189		СЛМ	1(2)	
0190		ПВ	ACCST(7)	
0191		СЧ	DRUM	
0192		ЗП	INFOR	
0193	WRITE	ОБМЕН	INFOR	
0194		СЧ	INFOR	





