

12

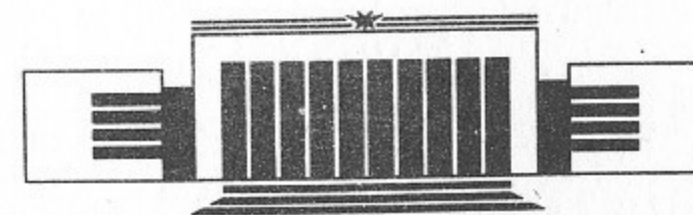
ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ СО АН СССР



В.Р. Козак

**МАТОБЕСПЕЧЕНИЕ  
ДЛЯ ЭВМ ОДРА И ОДРЕНОК,  
ПРОГРАММЫ ОБРАБОТКИ  
ТЕКСТОВЫХ ФАЙЛОВ**

**ПРЕПРИНТ 88-23**



НОВОСИБИРСК

Матобеспечение для ЭВМ Одра и ОДРЕНОК,  
программы обработки текстовых файлов

*В.Р. Козак*

Институт ядерной физики  
630090, Новосибирск 90, СССР

А Н Н О Т А Ц И Я

В препринте описаны редактор текстов #EDIT, текстовый процессор #GIRL, программы печати текстов стандартным (#KISS) и произвольным (#PRIN) шрифтами, копировщик изображения с ЦДР-2 на графическую DZM-180 (#LDZM). Описана структура текстовых файлов на ЭВМ ОДРА И ОДРЕНОК, особенности построения описываемых программ. В приложении приводятся описания всех программ, предназначенные для пользователя.

© Институт ядерной физики СО АН СССР

ВВЕДЕНИЕ

В ИЯФ СО АН СССР, начиная с 1971 г., для управления большими экспериментальными физическими установками широко используются 24-разрядные ЭВМ серии ОДРА-1325 (производства ПНР), являющимися функциональными аналогами английских машин ICL-1900. Эти машины относятся к классу универсальных ЭВМ, оснащены развитым программным обеспечением, а по своим функциональным возможностям (большое адресное пространство, средства поддержки мультипрограммирования, повышенная точность) превосходят распространенные миниЭВМ. В использовании ЭВМ ОДРА-1325 в ИЯФ СО АН СССР накоплен большой опыт, создано обширное системное и прикладное программное обеспечение, имеется значительный контингент специалистов, знакомых с особенностями работы на них.

Ряд недостатков штатных операционных систем с точки зрения управления физическим комплексом потребовал разработки универсальной операционной системы, ориентированной на управление [1], транслятора с фортраноподобного проблемно ориентированного языка TRAN, редактора текста и ряда других системных программ. Широкое распространение ЭВМ типа ОДРА, а затем микроЭВМ ОДРЕНОК [2] в качестве рабочих станций экспериментаторов, инженеров-разработчиков потребовали создания программного обеспечения, облегчающего такие работы, как тестирование аппаратуры, создание и ведение баз данных, документирование разработок, написание статей и препринтов.

Ниже описывается ряд программ, созданных в рамках описанных работ и относящихся к обработке текстов: редактор текстовых файлов, текстовый процессор, распечатчики текстовых файлов стандартным шрифтом и программируемым шрифтом, а также копировщик изображения графического дисплея на графический принтер.

### РЕДАКТОР ТЕКСТОВ

Широкое распространение мини- и микроЭВМ в качестве персональных рабочих станций и возрастание их вычислительной мощности привели к появлению редакторов экранного типа, существенно облегчающих работу пользователей. Такие редакторы были созданы в различных организациях для различных типов ЭВМ [3, 4]. Экранное редактирование базируется на принципе перемещения экрана алфавитно-цифрового дисплея над текстовым файлом и редактирования видимых участков текста по командам, вводимым с функциональной клавиатуры. В процессе редактирования пользователь устанавливает высвечиваемый указатель (курсор) в определенное место экрана и командами с клавиатуры заменяет, вставляет или удаляет символ или строку. Следует отметить, что работы по созданию редакторов на ЭВМ типа ОДРА ведутся довольно давно [5]. Ниже описывается последняя версия редактора #EDIT.

Хронологически первыми терминалами на ЭВМ ОДРА были терминалы VT-340, поэтому редактор ориентирован на редакторские функции этого дисплея. Принята работа с текстом, содержащимся в программном буфере редактора. Предельный размер отдельного текстового файла ограничен максимальным размером программы за вычетом размера собственно редактора, т. е. около 110 кбайт. Это ограничение, однако, не создает больших неудобств, так как все программы, работающие с большими объемами текстов, могут обрабатывать несколько файлов последовательно.

Дисковые накопители, используемые на ЭВМ ОДРА и микроЭВМ ОДРЕНОК, эмулируют магнитный барабан, обеспечивающий запись и чтение блоков произвольной длины со словной адресацией. Текстовые файлы имеют рекордовую структуру. Первые два слова в файле — комментарий — признак файла. По соглашению первое слово текстовых файлов содержит константу «ТЕХТ», что и является признаком текстовых файлов, а второе слово — расшири-

тель имени. Рекордовая структура начинается с третьего слова. Рекорд включает в себя длину рекорда в словах и затем цепочку символов. Признаком конца файла является рекорд, состоящий из одного слова, содержащего #32323232 (\*\*\*\* в текстовом виде).

Представление текстовой информации в ICL-1900 несколько отличается от общепринятого. ЭВМ ОДРА обладает 6-разрядным байтом, а для кодировки стандартного набора символов (большие латинские, большие русские и управляющие символы) требуется не менее 7 разрядов. Применяется следующий механизм: кроме кодов символов, существуют три выделенных кода регистров. Коды #74, #75, #76 соответствуют кодам латинских букв, русских букв и спецсимволов. Когда в тексте встречаются коды #74 или #75, все последующие символы трактуются как латинские или русские буквы, соответственно. Каждая новая строка считается начинающейся с латинских букв, если первым байтом не является регистр #75. В отличие от русского и латинского регистров, код спецрегистра относится только к следующему байту. Таблица перевода кодировки ASCII в коды ICL приведена в Приложении.

Появление терминалов с расширенным шрифтом (большие и маленькие русские и латинские буквы) вызвало некоторые трудности, так как они не вписывались в существовавшие правила. Было принято решение использовать для смены шрифта стандартные спецсимволы CTRL-N и CTRL-O (коды ASCII #16 и #17, соответственно). Считается что строка начинается с больших букв. Таким образом, если в строке не встречаются коды смены шрифта, то она считается состоящей только из больших букв. Это обеспечило совместимость старых текстовых файлов и файлов с расширенным шрифтом.

Редактор текстовых файлов ориентирован прежде всего на написание текстов программ, поэтому текст рассматривается как набор строк, каждая из которых не связана с другими и обрабатывается независимо от них. Каждая строка имеет свой номер (строки нумеруются по порядку), который всегда выводится на экран в начале строки. Это вызвано широко распространенными манипуляциями программистов со строками (копирование, переход на другую строку и т. п.).

На экране терминала распечатывается номер строки, директивная зона и собственно строка текста. Директивная зона служит для ввода и исполнения наиболее употребительных команд редактора без выхода из экранного режима. Этими директивами являются: перемещение вверх/вниз по тексту, обновление кадра (когда

переписывается весь экран), директивы поиска цепочки знаков в тексте, вставка строк из другого места в тексте, чтение строк из другого файла на диске, переход на какую-либо строку и пр. Перемещение курсора вправо переводит редактор в режим редактирования строки. Здесь функционирует другой набор директив, более узкий. Кроме перемещения по тексту курсором, здесь действуют различная табуляция, расщепление строки, слияние строк, включение/выключение режима «автовставки», а также отмена редакции строки.

В последней версии редактора добавлена еще одна возможность: кроме директив, принимаемых непосредственно в строке и в директивной зоне, введен командный режим, который позволил реализовать ряд новых директив. В командном режиме возможны следующие функции: поиск и замена одной цепочки символов на другую как автоматически, так и по команде оператора, изменение размера экрана (24 или 16 строк), изменение позиций табуляторов, фиксирование размера буфера и пр. Таким образом, наиболее употребительный минимум директив выполняется в экранном режиме, а расширение возможностей редактора осуществляется в командном режиме.

При редактировании строки, она копируется во внутренний буфер и преобразуется в промежуточный код (в каждом слове лежит в двух байтах код символа с его регистром и признаком шрифта). Это существенно облегчает работу, так как после преобразования каждому символу на экране соответствует одно слово во внутреннем буфере, независимо от регистра и формата. Внутреннее представление строки используется как при редакции строки, так и при поиске цепочки символов и при замене одной цепочки на другую. После окончания редакции строки преобразуется в стандартный рекорд и, после устранения последних незначащих пробелов, замещает предыдущее содержимое редактируемой строки.

Подробное описание редактора приводится в Приложении.

## ТЕКСТОВЫЙ ПРОЦЕССОР

Распространение персональных рабочих станций привело к широкому использованию их для написания статей и препринтов, составления отчетов, описаний и другой документации. Возможности редактора не вполне устраивали пользователей, что инициировало разработку текстового процессора. Как правило, обработка тек-

стов бывает либо диалоговая, когда она совмещена с редактором и позволяет исполнять отдельные функции сразу после ввода директивы и сразу же видеть результат на экране терминала, или же функции обработки текстов возлагаются на специализированный текстовый процессор, управляемый директивами, содержащимися в самом тексте. Для начинающих пользователей, как правило, более удобным являются программы первого рода, так как они несложны в использовании и позволяют непосредственно после воздействия директивы на текст наблюдать ее результат. Однако, ограниченные ресурсы применяемых мини- и микроЭВМ, а также ограничения, присущие процессорам первого рода, стимулировали реализацию второго подхода.

Результатом разработки явилась программа, позволяющая форматировать текст по страницам, выравнивать строки, делать переносы и реализовать ряд других функций. Обращать несложные тексты можно, зная всего несколько директив.

Текстовый процессор может обрабатывать произвольное число файлов, формируя один или несколько выходных текстовых файлов, которые можно просмотреть стандартным редактором или распечатать программой-принтером. Функции процессора:

- форматирование строки;
- форматирование страницы;
- исполнение директив.

Формирование строк происходит следующим образом: процессор считывает очередную строку из входного файла, анализирует ее на наличие в ней директивы, затем проверяет не закончился ли блок строк, который начинается с абзаца (не менее 4 первых пробелов) и кончается либо следующим абзацем, либо пустой строкой, либо концом файла. Если блок не закончился, то процессор добавляет одну или несколько входных строк к внутреннему буферу. Если длина буфера превышает размер стандартной выходной строки, то формируется выходная строка и записывается в выходной файл. Если считан целый блок, то буфер разгружается полностью, остаток выделяется в отдельную укороченную строку. Если переносы слов были запрещены, то процессор формирует строку из столько слов, сколько вмещается в выходную строку, затем равномерно добавляет в текст пробелы до тех пор, пока длина строки не станет стандартной. Если же разрешены переносы, то процессор формирует строку из столько символов, сколько вмещается в стандартную строку. Если произошел разрыв слова, то процессор проверяет допустим ли перенос в этом месте, а если недопустим,

то формирует строку на один символ короче и снова проверяет допустимость переноса. Когда найдено место переноса или окажется целое число слов, строка разравнивается пробелами до стандартной длины и записывается в выходной файл.

Алгоритм переноса примитивен, но, как показал опыт эксплуатации программы, довольно эффективен. Процессор прежде всего рассматривает первую и вторую часть слова и считает в них количества гласных и согласных букв. Если хотя бы одно из них равно нулю, то перенос запрещен. Кроме гласных и согласных, введен третий тип знаков, таких как ь, й, :, ;, а также цифры и прочие служебные знаки, которые не входят в баланс гласных и согласных. Кроме этого, переносимая часть слова не должна начинаться со знаков третьего типа.

Если выполнен баланс гласных и согласных букв, то процессор применяет последовательно ряд дополнительных правил:

- если перенос между эквивалентными буквами — перенос разрешен;
- если первая часть слова заканчивается двойной буквой — переносить нельзя;
- если вторая часть слова начинается двойной буквой — переносить нельзя;
- если вторая часть слова начинается знаком третьего типа — переносить нельзя;
- если перенос предполагается между согласными, то в начале второй части слова не должно быть двух согласных подряд и т. д.

Кроме этого, есть возможность указать какое число знаков можно оставлять в левой части переносимого слова. Это позволяет регулировать количество переносов на странице.

Страницы формируются с использованием нескольких директив. Этот механизм может быть включен и выключен в тексте, а перевод на следующую страницу (завершение текущей) может осуществляться как по ее заполнению, так и безусловной командой, а также по условию если до конца текущей страницы осталось указанное число строк. Условный перевод страницы очень удобен, так как в противном случае бывают казусы с началом **новой** главы, в которой после заголовка на странице печатается две строки, относящиеся к этому заголовку, или вообще ни одной строки.

Директива в тексте предваряется знаками «# #», после которых следует директива. Если директива не содержит параметра, то анализируются первые 4 знака, если содержит параметр, то два

знака интерпретируются как директива, а последующие 2-3 знака как числовой параметр.

Для распечатки произвольными переключаемыми шрифтами, текстовый процессор может переключаться в соответствующий режим, в котором пары символов, начинающиеся с «^», трактуются как управляющие коды для распечатчика и не включаются в длину строки. Этот механизм описан в описании программы # PRIN. Подробное описание текстового процессора приводится в Приложении.

## РАСПЕЧАТКА ТЕКСТОВ

Пользователям ЭВМ приходится довольно часто распечатывать как тексты программ, так и описания, документацию и пр. Программа печати должна обладать рядом противоречивых возможностей: текст программ должен печататься с номерами строк, описания же выигрывают от их отсутствия, тексты с длинными строками разумно печатать в одну колонку, при распечатке же обычных текстов при этом пропадает половина формата бумаги и т. д. Программа, описываемая ниже, **удовлетворяет** большинству требований пользователей.

Программа построена по следующему принципу. После загрузки она спрашивает адрес DZM (номер терминального канала), имя файла и номер страницы, с которой нужно начинать печатать. Многим пользователям никаких директив вводить не нужно. Если же пользователь желает что-либо указать распечатчику, то он переводит его в режим приема директив. Для упрощения пользования программой, она предлагает возможные функции и подсказывает возможные ответы. Функции предлагаются в порядке частоты использования, т. е. наиболее употребительные предлагаются первыми. В любом месте можно отказаться от дальнейшего ввода директив и перейти к вводу имени файла.

Возможны следующие функции.

Распечатка на стандартном формате в одну колонку, в две одинаковые колонки (дублирование) и в две страницы (каждая колонка — отдельная страница).

Строки нумеруются и в конце страницы печатается дополнительная информация (имя файла, номер страницы, дата и время распечатки) или же на странице печатается только текст.

Можно также указать количество строк, которое нужно печатать на каждой странице, размер страницы в строках (это требуется для автоматической разбивки по страницам), можно автоматически нумеровать только страницы. Программа может печатать как отдельные фрагменты, так и весь текстовый файл.

Использование в последнее время расширенного алфавита при недостатке соответствующих терминалов привело к появлению файлов, в которых латинские буквы трактуются как русские заглавные, а русские буквы трактуются как маленькие. По директиве оператора распечатчик осуществляет автоматическую перекодировку текста при печати.

Распечатчик может быть запущен другой программой с передачей ему всей необходимой информации через специальный файл на диске. Если все параметры указаны правильно, то распечатчик выполняет все указанные функции и автоматически удаляет себя из памяти машины, причем может загрузить другую программу, указанную ему в списке параметров. Если же в файле указаны неправильные параметры или распечатчик не может обработать какую-либо ситуацию (например нет указанного файла или произошел сбой при его открытии), программа просит ввести дополнительную информацию с терминала.

Краткое описание директив распечатчика текстовых файлов приводится в Приложении.

#### РАСПЕЧАТКА ТЕКСТОВ ПРОИЗВОЛЬНЫМ ШРИФТОМ

При подготовке сложных описаний или статей с формулами часто требуется возможность печатать более читаемым шрифтом (обычный шрифт DZM-180 неудобочитаем, особенно маленькие буквы), а также вставлять в текст символы, которых стандартный знакогенератор не имеет. Для этой цели была разработана специальная программа, которая использует предварительно составленный знакогенератор из файла на диске и печатает на принтере в графическом режиме. Шрифты составляются программой #CHAR и хранятся на диске в сложном файле .ALPH. В одном сложном файле могут быть несколько шрифтов с произвольными именами. При загрузке #PRIN спрашивает имя шрифта, а затем просит ввести параметры, аналогичные параметрам для #KISS и #D100. Шрифт состоит из 512 символов. Младшие 256 символов

соответствуют кодировке графической DZM-180). Шрифт разбит на группы по 128 символов. Группа 0—коды КОИ-7; группа 1—коды маленьких букв; группы 2 и 3—дополнительные коды. В тексте можно произвести переключение на любую группу. Кроме смены группы, в тексте можно включить/выключить подчеркивание и инверсию цвета (негативное изображение). Это производится сочетанием «^0», где вместо нуля может стоять любой символ. Два таких символа приводят к тому, что печатается он сам, а любой другой символ приводит к следующим действиям: младшие 2 разряда переключают группу символов; 3-й разряд включает подчеркивание, 4-й разряд включает инверсию. Подчеркивание можно в определенных пределах изменять, оно хранится как знак с кодом #777.

Ниже приведены младшие 15 комбинаций и их действие:

0	группа 0	большие буквы
1	группа 1	маленькие буквы
2	группа 2	греческие буквы и дополнительные знаки
3	группа 3	резервная группа
4	группа 0	большие буквы + подчеркивание
5	группа 1	маленькие буквы + подчеркивание
6	группа 2	греческие буквы и дополнительные знаки + + подчеркивание
7	группа 3	резервная группа + инверсия
8	группа 0	большие буквы + инверсия
9	группа 1	маленькие буквы + инверсия
:	группа 2	греческие буквы и дополнительные знаки + + инверсия
;	группа 3	резервная группа + инверсия
<	группа 0	большие буквы + подчеркивание + инверсия
=	группа 1	маленькие буквы + подчеркивание + инверсия
>	группа 2	греческие буквы и дополнительные знаки + + подчеркивание + инверсия
?	группа 3	резервная группа + подчеркивание + инверсия

Кодировка знаков больших и маленьких букв соответствует кодировке терминалов, греческие буквы соответствуют своим аналогам из латинского шрифта, а математические и другие символы по возможности располагались для более удобного запоминания. Коды и символы приведены в Приложении.

## КОПИРОВАНИЕ ИЗОБРАЖЕНИЯ С ГРАФИЧЕСКОГО ДИСПЛЕЯ

В Институте широкое распространение получили графические дисплеи типа ЦДР-2. Очень часто графическую информацию с экрана дисплея желательно получить в виде твердой копии. Для этого была разработана программа, учитывающая различные способы включения в систему дисплея, графического принтера-плоттера DZM-180, а также реализующая некоторый ряд функций, удобных для пользователя. Программа с точки зрения пользователя устроена аналогично программе распечатчика, т. е. диалоговым образом, причем вопросы расположены в порядке употребительности данных функций. Для большинства пользователей диалог может включать в себя только определение адреса и позиции дисплея и указание формата распечатки. По желанию пользователя можно определять способ подключения к системе принтера-плоттера и его адрес, цвет считываемого изображения, размер экрана, подлежащего копированию и т. д. Краткое описание директив копировщика приводится в Приложении.

## ПРИЛОЖЕНИЕ

### Перевод кодов ASCII в коды ICL

Код регистра	ASCII	ICL		ASCII	ICL		ASCII	ICL		ASCII	ICL
		# 76			# 74 или # 75			# 74			# 75
	000	20	Пробел	040	20	@	100	40	Ю	140	40
	001	21	:	041	21	A	101	41	А	141	41
STX	002	22	"	042	22	B	102	42	Б	142	42
ETX	003	23	#	043	23	C	103	43	Ц	143	43
	004	24	\$	044	24	D	104	44	Д	144	44
	005	25	%	045	25	E	105	45	Е	145	45
	006	26	&	046	26	F	106	46	Ф	146	46
	007	27	'	047	27	G	107	47	Г	147	47
Курс. влево	010	30	(	050	30	H	110	50	Х	150	50
TAB	011	31	)	051	31	I	111	51	И	151	51
LF	012	32	*	052	32	J	112	52	Й	152	52
IL	013	33	+	053	33	K	113	53	К	153	53
HOME	014	34	,	054	34	L	114	54	Л	154	54
RETURN	015	35	—	055	35	M	115	55	М	155	55
SO	016	36	.	056	36	N	116	56	Н	156	56
SI	017	37	/	057	37	O	117	57	О	157	57
	020	00	0	060	00	P	120	60	П	160	60
	021	1	1	061	01	Q	121	61	Я	161	61
	022	02	2	062	02	R	122	62	Р	162	62
	023	03	3	063	03	S	123	63	С	163	63
	024	04	4	064	04	T	124	64	Т	164	64
	025	05	5	065	05	U	125	65	У	165	65
	026	06	6	066	06	V	126	66	Ж	166	66
	027	7	7	067	07	W	127	67	В	167	67
Курс. вправо	030	10	8	070	10	X	130	70	Ь	170	70
Курс. вверх	031	11	9	071	11	Y	131	71	Ы	171	71
Курсор вниз	032	12	:	072	12	Z	132	72	Э	172	72
	033	13	;	073	13	[	133	73	Ш	173	73
IC	034	14	<	074	14	\	134	7664	Э	174	7670
DC	035	15	=	075	15	]	135	7665	Щ	175	7671
DL	036	16	>	076	16	^	136	7666	Ч	176	7672
ERASE	037	17	?	077	17	—	137	7667	DEL	177	7673

ЗНАКОГЕНЕРАТОР 1000 ДЛЯ ПРОГРАММЫ #PRIN

КОД СИМВОЛ	КОД СИМВОЛ	КОД СИМВОЛ	КОД СИМВОЛ
# 40	#240	#440	#640
# 41	#241	#441 J	#641 l
# 42 "	#242 "	#442 e	#642 †
# 43 #	#243 #	#443 ↓	#643 †
# 44 x	#244 x	#444 ~	#644 e
# 45 %	#245 %	#445 ←	#645 →
# 46 &	#246 &	#446 e	#646 z
# 47 ' .	#247 ' .	#447 ^	#647 v
# 50 (	#250 (	#450 €	#650 n
# 51 )	#251 )	#451 e	#651 U
# 52 *	#252 *	#452 e	#652 e
# 53 +	#253 +	#453 ±	#653 e
# 54 ,	#254 ,	#454 ^	#654 v
# 55 -	#255 -	#455 +	#655
# 56 .	#256 .	#456 ...	#656 e
# 57 /	#257 /	#457 #	#657 #
# 60 0	#260 0	#460 o	#660 o
# 61 1	#261 1	#461 1	#661 1
# 62 2	#262 2	#462 2	#662 2
# 63 3	#263 3	#463 3	#663 3
# 64 4	#264 4	#464 4	#664 4
# 65 5	#265 5	#465 5	#665 5
# 66 6	#266 6	#466 6	#666 6
# 67 7	#267 7	#467 7	#667 7
# 70 8	#270 8	#470 8	#670 8
# 71 9	#271 9	#471 9	#671 9
# 72 :	#272 :	#472 ::	#672 ::
# 73 ;	#273 ;	#473 ^	#673 v
# 74 <	#274 <	#474 <	#674 «
# 75 =	#275 =	#475 =	#675 ≈
# 76 >	#276 >	#476 >	#676 »
# 77 ?	#277 ?	#477 v	#677 ≈

#500 @	#520 П	#540 @	#560 π
#501 A	#521 В	#541 α	#561 ϑ
#502 B	#522 P	#542 β	#562 ρ
#503 H	#523 Σ	#543 η	#563 ϑ
#504 Δ	#524 T	#544 δ	#564 τ
#505 E	#525 Y	#545 ε	#565 υ
#506 φ	#526 Y	#546 φ	#566 Э
#507 Γ	#527 Ω	#547 γ	#567 ω
#510 X	#530 E	#550 x	#570 ξ
#511 I	#531 Ψ	#551 l	#571 ψ
#512 \$	#532 Z	#552 №	#572 ζ
#513 K	#533 III	#553 и	#573 #
#514 Λ	#534 II	#554 λ	#574 L
#515 M	#535 J	#555 μ	#575 #
#516 N	#536 II	#556 ν	#576 #
#517 O	#537	#557 o	#577

Эта таблица поможет ориентироваться в группах

" # x % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?	Группа 0
@ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _	
Ю А В Ц Д Е Ф Г Х И Й К Л М Н О П Я Р С Т У Ж В Ъ Ы З Ш Э Щ Ч	
" # x % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ?	Группа 1
` a b c d e f g h i j k l m n o p q r s t u v w x y z {   }	
ю а б ц д е ф г х и й к л м н о п я р с т у ж в ъ ы з ш э щ ч	
] e ↓ ~ ← ~ ^ € ∃ ⊕ ± ^ ÷ ... # 0 1 2 3 4 5 6 7 8 9 : ; < = > v	Группа 2
@ A B H Δ E F Γ X I \$ K A M N O П B P Σ T Y Y Ω E Ψ Z III J II	
@ α β η δ ε φ γ χ ι № λ μ ν ο π ϑ ρ σ τ υ ε ω ξ ψ ζ # 1 # #	
l † † ∞ → z √ n U ⊕ ⊙ ⊙ # 0 1 2 3 4 5 6 7 8 9 ≈ ∇ « ≈ » ≈	Группа 3

Примеры использования спецзнаков:  
 $H_2SO_4$  - серная кислота.  
 $\int \sin(x) dx = \cos(x) + const$



# ЗНАКОГЕНЕРАТОР 2000 ДЛЯ #PRIN

```
!"#%&'()*+,-./0123456789:;<=>?  
@ABCDEFGHIJKLMN OPQRSTUVWXYZ [\ ] ^ _  
ЮАБЦДЕФГХИЙКЛМНОПЯРСТУЖВЬЫЗШЭЩЧ  
!"#%&'()*+,-./0123456789:;<=>?  
`abcdefghijklmnopqrstuvwxyz{|}  
юабцдефгхийк лмнопярсту жвьыизшэщч  
[↔] ↓ ~←↘ ↙ ↞ ⊕ ± / ÷ ... ≠ 0 1 2 3 4 5 6 7 8 9 :: Δ < ≡ > √  
© ABH Δ E Φ Γ X I $ K Λ Μ Ν Ο Π Θ Ρ Σ Τ Υ V Ω Ξ Ψ Ζ Η Ι Ι  
Ⓢ α β η δ ε φ γ χ ι λ μ ν ο π ϑ ρ σ τ υ ε ω ξ ψ ϙ Ϡ ϡ Ϣ ϣ  
| ↑ ∞ → → √ ∪ ∅ ⊕ / ⊙ ⊚ ⊛ ⊜ ⊝ ⊞ ⊟ ⊠ ⊡ ⊢ ⊣ ⊤ ⊥ ⊦ ⊧ ⊨ ⊩ ⊪ ⊫ ⊬ ⊭ ⊮ ⊯ ⊰ ⊱ ⊲ ⊳ ⊴ ⊵ ⊶ ⊷ ⊸ ⊹ ⊺ ⊻ ⊼ ⊽ ⊾ ⊿
```

$H_2SO_4$  — **серная кислота**  
 $\int \sin(x) dx = -\cos(x) + const$

## ОПИСАНИЕ РЕДАКТОРА #EDIT

Редактор текстовых файлов предназначен для написания и редактирования программ и различных текстов. Редактор #EDIT ориентирован на работу с терминалом типа VT-340. Информация из файла помещается в буфер программы и, по окончании редакции, снова записывается на диск в прежний файл. Длина текстового файла может быть не более 28000 слов. Редактор изменяет свой размер в зависимости от размера текстового файла.

#EDIT позволяет пользоваться расширенным шрифтом (большие и маленькие русские и латинские буквы), имеет динамическое изменение внутреннего буфера, имеет командный режим, допускает большую длину рекордов (до 100 слов).

При использовании расширенного шрифта кодировка смены шрифта производится спецсимволами CTRL-N (перевод в режим маленьких букв) и CTRL-O (перевод в режим больших букв), как в некоторых терминалах и DZM-180. Для совместимости старых файлов с новыми принято, что при отсутствии в строке управляющих спецсимволов все знаки трактуются большими.

### Начало редактирования

При запуске редактора непосредственно  
LG EDIT 20  
или с помощью загрузчика  
LG VVVV 20  
или директивой системы  
ED

редактор представляется на экране терминала и просит указать имя файла, выведя текст:

РЕДАКТИРОВАНИЕ ФАЙЛА <—

При ответе <;> или <ESC> #EDIT уничтожает себя в оперативной памяти машины.

При ответе </> #EDIT уничтожает себя в оперативной памяти машины, загружает в память ЭВМ транслятор #TRAN и запускает его с 20-й ячейки.

Загрузчик — это обычная программа минимальной длины (1 блок). Функцией загрузчика является загрузка редактора на терминал с изменением имени редактора. Заменяется последний символ его имени на номер терминала.

Для того чтобы отредактировать уже имеющийся на диске текстовый файл, необходимо ввести его имя (четыре латинские буквы) и закончить ввод пробелом. При вводе более длинного имени редактор лишние знаки проигнорирует, а первые четыре примет за имя файла.

После этого #EDIT проверяет наличие файла на диске, определяет его длину, при необходимости изменяет свой размер и (если ему это удалось) читает файл и проверяет его рекордную структуру. При отсутствии ошибок, редактор перезапускает себя с 21-й ячейки. При этом на экране терминала распечатывается первая строка текста из файла и курсор помещается в директивную зону. Если в процессе перечисленных проверок были обнаружены ошибки, или файл слишком велик, редактор выводит соответствующие диагностические сообщения на экран терминала.

Если необходимо создать новый текстовый файл, нужно ввести его имя и комментарий и закончить ввод звездочкой или <LF>. Первые четыре байта трактуются как имя файла, следующие четыре байта — как комментарий, а остальные знаки игнорируются. #EDIT проверяет, что файла с таким именем на диске нет, и переходит к редактированию текста, образовав в буфере одну пустую строку. Физически файл будет образован после директивы <E>. Длина файла будет на 400 слов больше, чем использовано, но не может превышать длины буфера.

### Редактор строки

В отличие от ранних версий редактора, #EDIT любой текст принимает с помощью единого редактора строки, что позволило реализовать редакторские функции при вводе директив, имени файла и т. п. Единый редактор строки, принимая символы с экрана, анализирует их и все ненужные фильтрует (например, символ в спецрегистре, который не является терминатором, или если при приеме числа пользователь пытается ввести букву). Редактор строки в #EDIT может работать в трех режимах:

- 1) редактирование строки файла;
- 2) прием имени файла или цепочки знаков для поиска ее в тексте;
- 3) прием числа.

Во втором случае ограничена длина текста (16 позиций) и нет табуляции.

В третьем случае ограничена длина текста (5 позиций), нет та-

буляции и игнорируется попытка ввести не цифры.

Во всех трех режимах возможны все редакторские возможности: использование <IC>, <DC>, работа курсором, <CONTROL-F>, <CONTROL-S>, включение, выключение и использование режима автовставки и возможность закончить ввод в любой позиции строки.

### Директивная зона

Исходное положение курсора — в директивной зоне. На экране строка выглядит следующим образом:

```
1021 :20 CALL SAMAC(15,1Z)
```

Сначала печатается четырехзначный порядковый номер строки в файле (этот номер нигде не запоминается и печатается только для облегчения ориентации в тексте).

Далее следует директивная зона, из которой действует большинство директив, а затем печатается содержимое строки. Ниже описываются директивы, действующие из директивной зоны.

<Пробел> и <курсор вниз> вызывают перемещение курсора на следующую строку вниз в пределах директивной зоны. Опуститься ниже последней строки невозможно.

<Курсор влево> и <курсор вверх> вызывают перемещение курсора на следующую строку вверх в пределах директивной зоны. Подняться выше первой строки невозможно.

<ERASE> и <HOME> — восстановление кадра. Редактор распечатывает на экране терминала 23 (15) строк около текущей строки.

Цифра 1 ÷ 9 — переход на 10 ÷ 90 строк вниз, после этого восстановление кадра. При переходе на несуществующую строку курсор окажется на последней строке.

<—> действует только перед вводом цифры, совместно с ней переход на 10 ÷ 90 строк вверх, после этого восстановление кадра. При переходе на несуществующую строку курсор окажется на первой строке.

<P> — переход на строку, номер которой редактор просит ввести. Если ничего не ввели (сразу нажат пробел), курсор останется на прежней строке. При попытке ввести нулевой или несуществующий номер курсор перейдет на первую или последнюю строку, соответственно. Если строка существует, то она распечатается на месте предыдущей, если же указана несуществующая строка, то #EDIT восстановит кадр.

<I> — копирование строк из буфера. Возможные формы ввода:

I 12,4 — скопировать 4 строки, начиная с 12-й.

I 12—17 — скопировать строки с 12-й по 17-ю.

I 12 — скопировать 12-ю строку.

Скопированные строки вставляются перед текущей строкой. Возможна многократная вставка одной строки. Для этого строку копируют в нужное место директивой I N1, затем устанавливают курсор на нее и директивой I NN,K копируют строку K раз. NN—номер текущей строки. Аналогично можно произвести многократное копирование нескольких строк.

⟨R⟩—копирование строк из файла на диске. Кроме имени файла, необходимо указать номера строк аналогично предыдущей директиве. Если указанного файла нет на диске, он не текстовый, произошел сбой при чтении, нарушена рекордовая структура или указаны несуществующие строки, директива игнорируется.

⟨C⟩—поиск в тексте строки, в которой содержится определенная цепочка символов. Вводится цепочка любых символов (до символа ⟨ETX⟩) и редактор последовательно просматривает все строки, начиная со следующей, и если находит строку с такой же цепочкой знаков, то распечатывает эту строку и помещает курсор в ее директивную зону. Если нужная цепочка в тексте не найдена, то курсор возвращается на исходную строку. После последней строки начинает просматриваться первая. Клавиша ⟨RET⟩ аннулирует введенную цепочку. Заканчивать ввод можно в любой позиции. В отличие от предыдущих редакторов #EDIT производит поиск именно той последовательности знаков, которая была введена, т. е. если задан поиск маленьких латинских букв, то и большие и русские буквы он будет игнорировать. Во время поиска, курсор перемещается в левую позицию строки.

⟨S⟩—поиск в тексте строки, которая начинается с указанной цепочки знаков. Особенности те же, что и для предыдущей директивы.

⟨N⟩—повторяет поиск по директивам ⟨C⟩ или ⟨S⟩ без перебора цепочки символов. Это удобно, например, при просмотре каких-то переменных, COMMONов и т. п.

⟨DL⟩—устранение строки, в директивном поле которой стоит курсор. Если в тексте осталась единственная строка, то #EDIT ее не устраняет.

⟨Q⟩—отмена редакции. Редактор выходит на прием нового имени файла, не записывая отредактированную версию на диск. Если директива была введена ошибочно, то нужно нажать клавишу ⟨ERASE⟩, редактор восстановит кадр и курсор окажется на

той строке, на которой был введен символ ⟨Q⟩. Можно также перезапустить редактор с 21-й ячейки.

⟨E⟩—запись текста на диск и переход на прием нового имени файла.

⟨/⟩—запись текста на диск, редактор уничтожает себя в памяти ЭВМ и загружает транслятор #TRAN.

⟨A⟩—запись текста на диск, редактор уничтожает себя в памяти ЭВМ и загружает транслятор #XPLA.

⟨K⟩—запись текста на диск, редактор уничтожает себя в памяти ЭВМ и загружает распечатчик текстов #KISS.

⟨G⟩—запись текста на диск, редактор уничтожает себя в памяти ЭВМ и загружает текстовый процессор #GIRL.

⟨Курсор вправо⟩, ⟨CTRL-F⟩, ⟨CTRL-G⟩, ⟨TAB⟩—перемещение курсора вправо, перевод редактора в режим редактирования строки.

⟨LF⟩—образование строки ниже текущей и выход в редактирование ее.

⟨IL⟩—образование строки выше текущей и выход в редактирование ее.

⟨ETX⟩—переход в командный режим.

### Редакция строки

После выхода на строку (или после директив ⟨LF⟩, ⟨IL⟩) редактор переходит в режим редактирования строки. Здесь можно пользоваться следующими директивами.

⟨Курсор влево⟩, ⟨курсор вправо⟩, ⟨IC⟩, ⟨DC⟩ работают также, как и на терминале типа VT-340 в OFF-LINE.

⟨ETX⟩—конец редакции строки, курсор перемещается в директивную зону.

⟨LF⟩—конец редакции строки, редактор образует новую строку ниже отредактированной.

⟨IL⟩—конец редакции строки, редактор образует новую строку выше отредактированной.

⟨Курсор вверх⟩—конец редакции строки, курсор перемещается на строку вверх на ту же позицию. На первой строке директива игнорируется.

⟨Курсор вниз⟩—конец редакции строки, курсор перемещается на строку вниз на ту же позицию. На последней строке директива игнорируется.

⟨ERASE⟩, ⟨HOME⟩, ⟨RETURN⟩—отмена редакции строки.

Если эта строка только что вводилась, она уничтожается; если корректировалась имевшаяся строка, редактор восстанавливает прежний вариант и переписывает его на экране.

<CTRL-F> — перемещение курсора на 8 позиций вправо.

<CTRL-S> — перемещение курсора на 8 позиций влево. Если курсор переместился левее левой позиции, переход в директивную зону.

<CTRL-Q> работает следующим образом: если правее курсора есть текст, то из него образуется новая строка, которая помещается ниже текущей, а в текущей строке остается то, что было левее курсора. Если же правее курсора, кроме пробелов, ничего нет, то нижняя строка перемещается в текущую, начиная от позиции курсора. В обоих случаях после выполнения директивы курсор перемещается в директивную зону.

<CTRL-G> — перемещение вправо к началу следующего слова, т. е. первому символу после одного или нескольких пробелов. После последнего слова курсор перемещается в конец строки.

<CTRL-T> — перемещение влево к началу предыдущего слова.

<CTRL-A> — включение режима «автовставки».

<CTRL-E> — выключение режима «автовставки».

При включенном режиме «автовставки» ввод каждого символа сопровождается предварительным исполнением функции «IS», т. е. текст строки раздвигается, освобождая место для вводимого текста.

<TAB> — перемещение курсора на следующую позицию табулятора-2.

Редактор имеет два программируемых табулятора. В командном режиме можно указать до десяти позиций табуляции каждому из табуляторов. Табулятор-1 работает следующим образом: по нажатию клавиши <ПРОБЕЛ>, курсор перемещается до следующей позиции табулятора, стирая все символы на своем пути. Если в момент нажатия <ПРОБЕЛ> курсор находился правее самой старшей позиции табулятора-1, то знак интерпретируется как обычный символ. Табулятор-2 работает несколько иначе: по нажатию клавиши <TAB>, курсор перемещается до следующей позиции табулятора, не стирая символов на своем пути. Если в момент нажатия <TAB> курсор находился правее самой старшей позиции табулятора-2, то <TAB> игнорируется.

<CTRL-N> — перевод в режим маленьких букв.

<CTRL-O> — перевод в режим больших букв.

## Командный режим

По <ETX> в директивной зоне или при приеме имени файла, редактор переходит в командный режим. Здесь существуют следующие директивы.

<ETX> — возврат из командного режима.

VT — размер экрана терминала 16 строк.

MERA — размер экрана терминала 24 строки.

TEXT — работа в режиме только больших букв, коды смены формата шрифта игнорируются.

ABCD — работа в режиме расширенного шрифта.

TRAN — ввод в табулятор-1 значения 6, удобного при написании фортрановских текстов.

PLAN — ввод в табулятор-1 значений удобных при написании текстов на языке ассемблера (6, 12, 15, 24).

TABL — ввод значений в табулятор-1.

TABC — ввод значений в табулятор-2.

Ввод значений табуляторов производится следующим образом. После приема директивы редактор распечатывает содержимое табуляторной памяти и переходит на прием позиций табулятора. Числа должны вводиться по порядку, от меньших к большим. Если в процессе ввода нажать <ETX> или ввести неправильные значения, редактор сообщит об этом и оставит прежние значения табулятора. Числа разделяются пробелами, пробел без ввода числа служит признаком конца ввода, после этого редактор изменяет значения табуляции. Если не было введено ни одного числа, то позиций табуляции не будет вообще.

FLOAT — разрешение редактору на динамическое изменение буфера.

FIX — фиксирование определенного размера буфера редактора (и его размера).

REPL — замена одной цепочки символов в тексте на другую. Пользователь вводит обе цепочки символов и зону действия директивы (в строках).

REPS — то же, что и REPL, но редактор, найдя подходящую для замены цепочку символов, заменяет ее только по согласию пользователя (символом Y).

REPA — то же, что и REPS, но зона действия директивы — весь текст.

Для первых двух директив требуется ввести зону действия в строках в стандартном виде (NN,K или NN-KK). После этого ре-

дактор принимает две цепочки символов, каждую до <ETX>. Первая цепочка ищется в тексте и заменяется на вторую. Символ <RET> при вводе цепочек трактуется как отказ от директивы. Для директивы REPA зону действия указывать не требуется.

Для директив REPS и REPA редактор заменяет одно сочетание на другое только по указанию пользователя. Найдя соответствующую цепочку в тексте, редактор распечатывает строку, помечает ее звездочкой (строкой ниже) и ожидает одну из возможных директив.

По <Y> редактор заменяет одну цепочку на другую, перепечатывает измененную строку и продолжает поиск в тексте следующей строки с нужной цепочкой.

По <ПРОБЕЛУ> редактор продолжает поиск в тексте следующей строки с нужной цепочкой.

По </> редактор прекращает поиск цепочек и переходит на прием команды.

#### Примечания

Если редактору какой-то знак не понравится, он просто его проигнорирует.

Все, что вы написали, можно записать в другой файл. Для этого нужно пустить редактор с 27-й ячейки и ввести имя файла с комментарием. Редактор создаст файл с этим названием и запишет в него свой буфер.

Когда длина вашего текста превысит длину файла, редактор печатает НЕ ЛЕЗЕТ. Нажмите любую клавишу и он выйдет на прием директив. В подобном случае рекомендуется сделать <E>, а потом программой #ZJAA расширить файл.

Редактор имеет два формата шрифта: обычный (только большие буквы) и расширенный. При создании нового файла тип шрифта определяется по режиму работы (например, была введена команда ABCD). Тип файла кодируется в его первом слове, в котором должно быть слово TEXT или ABCD. Следует иметь в виду, что при чтении файла с диска редактор автоматически изменяет режим по типу шрифта.

В режиме TEXT редактор игнорирует коды смены шрифта, приходящие с клавиатуры, но не фильтрует их, если они есть в тексте.

## # GIRL—ТЕКСТОВЫЙ ПРОЦЕССОР

### Для начинающих пользователей

Начинающим пользователям #GIRL рекомендуется пользоваться программой следующим образом:

1. Указать имя выходного файла: W:NAME. Выходной файл должен быть открыт до того программой #ZJAA.

2. Если в программе встречаются таблицы или другой текст, который желательно копировать, не обрабатывая, эта часть текста должна быть обрамлена строчками: ##NOOP и ##PROC.

3. Входной текстовый файл должен быть оформлен по следующим правилам:

- а) в тексте не должно быть переносов;
- б) первые 4 пробела в строке трактуются как начало абзаца;
- в) при обработке первые пробелы строки сохраняются, все остальные приводятся к одному, а после переноса строка выравнивается увеличением пробелов.

4. Если вас не устраивает ширина текста, то ее можно изменить директивой программе W168 или строкой в тексте ##W168. Рекомендуемая ширина строки: 60—70.

### Некоторые сведения о #GIRL

#GIRL работает с входными строками, которые могут трактоваться как директива либо как строка текста. При вводе с VT строка всегда трактуется как директива. Поэтому при вводе с VT #GIRL спрашивает:

КОМАНДНАЯ СТРОКА <—

В тексте строка с директивой всегда должна начинаться с сочетания ##. Следующие позиции трактуются как директива. При вводе директивы с VT сочетание ## вводить не нужно.

### Директивы

Директивы бывают простыми (END, BUFO, CLEA, CARR...) и сложными (PR12, LP66, W164, SH4...).

Простая директива либо взводит, либо сбрасывает ключ в программе. Такие директивы распознаются по первым четырем знакам, остальные знаки игнорируются. Таким образом, CARR=CARRY, CLEA=CLEAR, COMM=COMMENT и пр.

Сложная директива состоит из собственно директивы и некоторого числа, которое можно опускать. Собственно директива состоит из двух букв (WI, NP, LF...). Число состоит не более чем из трех цифр. Программа, анализируя число, просматривает три знака либо до пробела.

При вводе директивы с терминала можно пользоваться редакторскими знаками: управление курсором, IC, DC. Ввод заканчивается пробелом в любой позиции строки (#GIRL помнит сколько знаков было введено). Знак <DEL> игнорируется.

При вводе директивы с терминала и при встрече ее в тексте #GIRL распознает и трактует ее одинаково. Если директива ей не нравится, то #GIRL сообщает об этом и выводит на VT как сообщение об этом, так и не понравившуюся ей строку.

### Форматирование

Основное назначение #GIRL — выравнивание текста и разбивка его по страницам. Для выравнивания текста #GIRL рассматривает абзац как целую строку. Для этого в программе содержится буфер, который заполняется по мере чтения входных строк и разгружается сразу же, как только его размер превышает величину выходной строки. Следует учитывать, что вследствие этого директивы типа SH??, WI77 и пр. начинают исполняться не с того места, которое указано во входном файле (остаток буфера разгружается после принятия и распознавания директивы). Буфер разгружается полностью в следующих случаях:

- 1) если следующая строка начинается не менее чем с четырех пробелов;
- 2) в конце входного файла;
- 3) при приеме директив END, BUFO, DELE, LF??, NP0.

Страница состоит из определенного числа строк текста и нескольких служебных строк: три строки вначале (первая может включать в себя номер страницы, остальные пустые) и три строки в конце страницы (пустые).

Управление разбивкой текста на страницы и нумерацией страниц осуществляется директивами PAGE, NOPA, PG0, PG77, NP0, NP60, LP60 (см. ниже).

Если разбивка на страницы отключена директивой NOPA, то служебные строки в тексте отсутствуют и нумерация страниц не производится. Того же эффекта можно добиться, указав LP999, хотя при этом будут введены первые три служебные строки.

Кроме вертикального размера, страница характеризуется и горизонтальным. Директива WI??? задает размер правого края текста, а SH??? — сдвиг начала текста.

### Файлы

#GIRL работает с входными файлами стандартного текстового формата.

Если выходным файлом указан VT, рекомендуется пользоваться директивой OV??, чтобы вывод приостанавливался через определенное число строк.

При указании выходного файла W:NAME файл открывается, переписываются слова комментария на TEXTGIRL и записывается признак конца текста (\*\*\*\*). При каждой записи одновременно записывается и новый «конец файла».

#GIRL может обрабатывать текст из нескольких файлов, последовательно. Для этого в конце каждого файла должна стоять директива ##R:NAME2.

### Переносы

Выходная строка формируется следующим образом: сначала #GIRL пытается сделать перенос (если он не запрещен), потом выравнивает длину строки пробелами. Если директивой PR?? установлена слишком большая зона защиты, то выравнивания строки может не получиться.

Если переносы запрещены, то при малой ширине выходной строки слово может оказаться длиннее строки. В таких случаях оно игнорируется (с соответствующим сообщением на VT).

### Описание директив

CLEA — чистка ключей, (почти эквивалентно рестарту с 20-й ячейки). Не чистится имя входного файла и текущий адрес в нем (иначе эту директиву упоминать в файле было бы невозможно). Поскольку #GIRL помнит все встретившиеся директивы, рекомендуется при обработке файлов либо перезапускать #GIRL с 20-й ячейки, либо первую директиву во входном файле делать ##CLEAR. Во втором случае имя выходного файла тоже должно указываться во входном файле.

W:NAME — записывать обработанный текст в файл NAME. Ес-

ли файл есть на диске, то он стирается (признак конца записывается в третьем слове) и переписываются слова комментария на TEXTGIRL. Счетчик выходных строк не обнуляется, т. е. разбивка на страницы продолжается.

W:NAMEABCD—записывать обработанный текст в файл NAME, 1-е слово файла будет ABCD—признак расширенного алфавита.

W:VT—выводить обработанный текст на VT.

R:NAME—читать текст из входного файла NAME.

R:VT—ввод директив с VT.

DELE—программа уничтожает себя в памяти машины. Эта директива может быть последней в файле, хотя и не обязательно. Кроме директивы DELE, #GIRL устраняет себя из памяти ЭВМ по спецзнаку </>.

BUFO—вывести остаток буфера в выходной файл. Далее обработка текста продолжается в соответствии с директивами и текстом входного файла.

END—вывести остаток буфера в выходной файл и закрыть файл. После этого #GIRL считает VIT входным и выходным файлами. Это бывает полезным при обработке не всего текста, а, например, первой страницы (подбор удобных параметров в WI??, SH?? ...

COMM—дальнейший текст игнорировать до директивы NOCO.

NOCO—отмена действия предыдущей директивы.

Текст, обрамленный этими директивами, игнорируется, включая директивы, поэтому, можно писать в этой зоне произвольные комментарии.

NOOP—последующий текст копировать без обработки. Встречающиеся при этом в тексте директивы #GIRL принимает и распознает.

PROC—отмена действия предыдущей директивы.

Эти директивы могут быть полезны при встрече в тексте таблиц и пр., т. е. таких текстов, которые необходимо откопировать, не обрабатывая. Страничный механизм продолжает функционировать, т. е. #GIRL может разорвать таблицу на две страницы.

CARR—разрешаются переносы.

NOSA—запрещаются переносы.

Если пользователю не нравятся переносы вообще или переносы #GIRL, то он может их запретить либо вообще, либо на часть текста.

SPAC—непервые пробелы в тексте приводятся к одному.

NOSP—чистка пробелов запрещена.

#GIRL все пробелы в тексте приводит к единичным (за исключением первых). Если пользователю необходимы большие пробельные места (например для вписывания греческих букв и пр.) то он может запретить этот механизм вообще либо на часть текста.

WI???—ширина выходного текста (количество колонок): допустимо от 11 до 120.

SH???—количество пробелов в начале строки: допустимо от 0 до 60, но  $(WI?? - SH??) > 10$ .

LF???—сделать ??? количество пустых строк.

PG???—нумеровать страницы, первый номер=???

PG0—не нумеровать страницы.

PR???—первые ??? колонки защищены от вставления пробелов (иногда это бывает полезно): допустимо от 0 до 999.

LP???—длина страницы = ??? строк (не считая служебных, пустых). Допустимо 10—400.

NP???—перевод страницы, если до конца текущей осталось менее ??? строк.

NP0—безусловный перевод страницы.

PRIN—используется тогда, когда текст предназначен для распечатки программой #PRIN и в нем используются переключения групп шрифтов.

NOPR—отменяет предыдущую директиву.

OV???—при выводе на VT через ??? строк будет приостановка вывода с вопросом: NEXT?. По знаку </>—прекращение обработки, по любому другому знаку—продолжение.

По умолчанию (при рестарте #GIRL с 20-й ячейки) исполняется следующий набор директив:

W:VT R:VT WI64 SH0 CARRY NOPAGE PG0 PROC OV0 PRO  
NOCOMMENT NOSP LP60 NOPR.

## РАСПЕЧАТЧИК ТЕКСТОВЫХ ФАЙЛОВ # KISS

### Входы

20 — основной вход в # KISS, прием имени файла.

21 — перезапуск # KISS сначала при сохранении всех введенных директив.

23 — режим «печатающей машинки».

24 — выход на ввод номера страницы (директивы для печати должны быть уже введены).

25 — ввод директив для печати.

28 — старт в режиме AUTO, все директивы должны быть записаны в файле %COM.

В любом месте программы, если ответить на ее вопрос <HOME>, то # KISS стирает экран и печатает: ENTRY < —. При вводе цифры (с пробелом) от 0 до 9 # KISS перезапускается с 20 — 29-й ячейки, соответственно. При ответе </>, # KISS уничтожает себя в памяти машины.

### Директивы

При выходе на прием директив, # KISS задает вопросы в следующей последовательности:

ПЕЧАТАТЬ ДВЕ СТРАНИЦЫ (D,N,V) ?

Возможные ответы (в латинском или русском регистре):

D — # KISS будет печатать на одном листе бумаги две страницы, начало текста — на левой, продолжение — на правой. Если в тексте встретятся длинные строки (длина больше 80 позиций, включая остаточные пробелы), они будут печататься с переносом.

N — # KISS будет печатать на одном листе бумаги одну страницу. В целях экономии бумаги этот режим печати рекомендуется использовать только при большом количестве длинных строк.

V — # KISS будет печатать на одном листе бумаги две одинаковые страницы. Этот режим удобен для печати двух экземпляров одновременно. Перенос строк так же, как и при директиве D.

СТРОКИ НУМЕРОВАТЬ (D,N) ?

Возможные ответы (в латинском или русском регистре):

D — # KISS будет нумеровать распечатываемый текст и подпи-

сывать каждую напечатанную страницу в виде:

ФАЙЛ «NAME/КОММ» СТР. N DATA AA/BB/CC ВРЕМЯ DD/EE/FF  
N — # KISS не будет нумеровать и подписывать распечатанный текст.

ПЕЧАТАТЬ СТРОК В СТРАНИЦЕ ?

Распечатываемый текст печатается по страницам. Нужно указать по сколько строк распечатывать на странице (не менее 3).

СТРОК В СТРАНИЦЕ У DZM ?

# KISS может автоматически распечатывать весь текст. Для этого необходимо знать, сколько строк может вместиться на странице. (Существуют DZM-180 у которых эта величина отличается от стандартного значения 72). Кроме этого, можно указывать малый размер страницы, когда маленький текст (например, объявление) нужно распечатать в нескольких экземплярах на 1 странице бумаги (не менее 5).

ЗАДЕРЖКА НА СТРОКУ (СЕК) ?

При нормальной распечатке общение с DZM-180 завешивает канал связи. Если установить задержку на строку равную 1—2 секундам (при двухколоночной распечатке), то скорость печати практически не изменится, а канал будет свободен. Это связано с тем, что программа после передачи строки будет останавливаться на то время, за которое DZM-180 разгружает свой буфер.

ШАГ ПЕЧАТИ (КОЛ-ВО ПУСТЫХ СТРОК) ?

При распечатке текстов иногда бывает удобным печатать строки не подряд, а с некоторым интервалом, пропуская, например, еще одну строку. При вводе «0» будет сплошная печать.

СТРАНИЦЫ НУМЕРОВАТЬ (D,N) ?

Эта директива доступна только при распечатке текстов без нумерации строк. При утвердительном ответе # KISS нумерует страницы (в верхней части).

При вводе всех директив одинаковым образом действуют следующие спецзнаки:

<ПРОБЕЛ> перейти к следующей директиве. После ввода последней — рестарт с 21-ячейки (на ввод файла).

</> перейти к предыдущей директиве. Перед первой — рестарт с 21-ячейки (на ввод файла).

<RETURN> рестарт с 21-й ячейки (на ввод файла).



<HOME> перезапуск программы с ячейки, указываемой после <HOME>.

По умолчанию (по пробелу и пр.) параметры сохраняются.

При запуске программы с 20-й ячейки исходные значения директив:

ПЕЧАТАТЬ ДВЕ СТРАНИЦЫ (D,N,V) ?	D
СТРОКИ НУМЕРОВАТЬ (D,N) ?	D
ПЕЧАТАТЬ СТРОК В СТРАНИЦЕ ?	60
СТРОК В СТРАНИЦЕ У DZM ?	72
ЗАДЕРЖКА НА СТРОКУ (СЕК) ?	0
ШАГ ПЕЧАТИ (КОЛ-ВО ПУСТЫХ СТРОК) ?	0
СТРАНИЦЫ НУМЕРОВАТЬ (D,N) ?	N

### Распечатка

При запуске программы с 20-й ячейки #KISS просит ввести имя файла фразой:

ИМЯ ФАЙЛА :

При ответе </> #KISS уничтожает себя в оперативной памяти машины.

При ответе <ETX> #KISS переходит на ввод директив.

При ответе <HOME> #KISS переходит на ввод ячейки для рестарта.

После ввода имени файла #KISS проверяет, текстовый ли файл, печатает слово из поля комментария (второе слово в файле), длину файла, и составляет таблицу адресов начал страниц. После этого #KISS просит указать номер страницы для вывода.

#KISS сама разбирается, есть ли в тексте расширенный алфавит. Если же вы хотите, чтобы при печати латинские буквы трактовались большими, а русские маленькими, то нужно ввод имени файла закончить восклицательным знаком.

Если ввести номер страницы с пробелом, то #KISS распечатает эту страницу (или пару страниц) и снова спросит номер страницы. По пробелу вывод продолжится далее (последовательно). После последней страницы в файле будет печататься снова первая (если нажать пробел).

Если ввести номер страницы и закончить ввод не пробелом, а восклицательным знаком, то #KISS будет выводить весь текст, начиная с указанной страницы. Таким образом для распечатки всего файла достаточно на вопрос: НОМЕР СТРАНИЦЫ—ответить: !.

### Режим «печатающей машинки»

При запуске с 23-й ячейки #KISS переходит в режим «печатающей машинки». При этом #KISS принимает с терминала строку текста (можно пользоваться курсором и знаками <IC>, <DC>) до символа <ETX>. Эта строка затем перепечатывается на терминале и печатается на DZM-180. Работа с диском при этом не происходит.

По символу <RETURN> #KISS уничтожает себя в оперативной памяти машины.

### Режим AUTO

При запуске с 28-й ячейки #KISS переходит в режим AUTO. В этом режиме все директивы для программы должны находиться на диске в файле «%COM», определенным образом оформленные.

Если на диске нет файла «%COM», или в нем испорчены ключевые слова (нет информации), или неправильные директивы, #KISS перезапускается с 20-й ячейки.

Если все директивы устраивают программу, она распечатывает файл без манипуляций оператора, с соответствии с указаниями в файле.

## ОПИСАНИЕ ПРОГРАММЫ # LDZM

# LDZM предназначена для считывания информации из ЦДР-2 и распечатки «твердых копий» на графических DZM-180, подключенных к терминальному каналу ЭВМ. Имеется возможность распечатывать весь экран и только часть его, считывать избранные цвета, работать с ЦДРом в системном крейте и в периферийном.

### Входы

20 — основной вход в # LDZM, прием параметров.

21 — перезапуск # LDZM сначала при сохранении всех введенных директив.

24 — выход на распечатку (директивы для печати должны быть уже введены).

### Директивы

При запуске # LDZM задает вопросы в следующей последовательности:

НОМЕР ТЕРМИНАЛА =

Нужно ввести номер терминального канала, к которому подключена DZM-180.

АДРЕС КРЕЙТА =

Нужно указать адрес крейта с ЦДР-2. Номер системного крейта — 0.

НОМЕР ЦДР =

Указывается позиция ЦДР в крейте.

ФОРМАТ =

Нужно указать формат, которым желательно распечатать изображение. Возможные форматы:

1 — изображение с ЦДР копируется на DZM-180 «точка в точку»;

2 — изображение с ЦДР копируется на DZM-180 с растяжкой в 2 раза по горизонтали.

3 — изображение с ЦДР копируется на DZM-180 с растяжкой в 2 раза по обоим координатам.

Если указать формат 0, то программа переходит в режим накопления трех независимых картинок с последующей их одновременной распечаткой.

ЦВЕТ =

Указывается номер считываемых цветов (1—7).

ПЕЧАТАТЬ ОТ СТРОКИ =

Начальная строка, считываемая с экрана. По умолчанию: 0.

ПЕЧАТАТЬ ДО СТРОКИ =

Конечная строка, считываемая с экрана. По умолчанию: 255.

При вводе всех директив одинаковым образом действуют следующие спецзнаки:

<Пробел> перейти к следующей директиве. После ввода последней — рестарт с 21-ячейки.

</> перейти к началу диалога, а если это первый вопрос диалога, то программа уничтожается.

<PRINT> переход в режим «печатающей машинки». Этот режим не отличается от аналогичного режима в # KISS и здесь не описывается.

<LF> # LDZM печатает <LF> на DZM-180 (переводит строку) и снова просит ввести параметр.

По умолчанию (по пробелу и пр.) параметры сохраняются.

При запуске программы с 20-й ячейки исходные значения:

АДРЕС DZM	=4
АДРЕС КРЕЙТА =	=0
НОМЕР ЦДР =	=14
ФОРМАТ =	=1
ЦВЕТ =	=7
ЗАДЕРЖКА НА СТРОКУ (СЕК)	=0
ПЕЧАТАТЬ ОТ СТРОКИ	=0
ПЕЧАТАТЬ ДО СТРОКИ	=255

### Режим накопления кадров

В этом режиме # LDZM может накопить 3 независимых картинки и вывести их на одной странице бумаги. Это экономит бумагу, а также облегчает сравнение многих «картинок» (на странице размещается 3×3 кадра). В этом режиме возможны следующие директивы:

C — очистка буфера программы.

R — чтение ЦДР в буфер, указывается кадр 0—2.

L — печать всего буфера на DZM-180.

S — остановка программы.

<LF> — печать пустой строки на принтере.  
(0—9) — регулировка расстояния между кадрами при печати.  
</> — переход в обычный режим.

#### ЛИТЕРАТУРА

1. Алешаев А.Н., Белов С.Д., Левичев Б.В., Протопопов И.Я. Операционная система ЭВМ ОДРА для управления электрофизическими установками в ИЯФ СО АН СССР — Препринт ИЯФ СО АН СССР 80-194. Новосибирск, 1980.
2. Пискунов Г.С. Распределенная система управления ускорительно-накопительным комплексом ВЭПП-4 (структура и средства). Кандидатская диссертация ИЯФ СО АН СССР. Новосибирск, 1985 г.
3. Кекеев Ю.З., Талныкин З.А., Яковенко Н.С. Текстовый редактор на базе алфавитно-цифрового дисплея. — Автометрия, 1975, № 2, с.106—112.
4. Шувалов Б.Н. МИСС — многотерминальная система с разделением времени для подготовки и редактирования текстов. — Препринт ИЯФ СО АН СССР, 78-60. Новосибирск, 1987.
5. Храпкин П.Л. Редактор текстов. Работы молодых специалистов, выполненные в ИЯФ СО АН СССР в 1977—1978 годах. Отчет, Новосибирск, 1978.

В.Р. Козак

#### Матобеспечение для ЭВМ Одра и ОДРЕНОК, программы обработки текстовых файлов

Ответственный за выпуск С.Г.Попов

---

Работа поступила 9 февраля 1988 г.  
Подписано в печать 15.02. 1988 г. МН 08098  
Формат бумаги 60×90 1/16 Объем 2,5 печ.л., 2,0 уч.-изд.л.  
Тираж 290 экз. Бесплатно. Заказ № 23

---

Набрано в автоматизированной системе на базе фото-наборного автомата ФА1000 и ЭВМ «Электроника» и отпечатано на ротапинтере Института ядерной физики СО АН СССР,  
Новосибирск, 630090, пр. академика Лаврентьева, 11.