

У8405

К-174



Калмыкова Л.А.

Б1-10-83-722

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

1282/84

Б1-10-83-722

ДЕПОНИРОВАННАЯ ПУБЛИКАЦИЯ

Дубна 1983

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ
Лаборатория вычислительной техники и автоматизации

Ц 840.8
К-174

51-10-83-722

Л.А. Калмыкова

РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ РЕДАКТОРА
EDNA НА ЭВМ CDC-6500

ОБЪЕДИНЕННЫЙ ИНСТИТУТ
ЯДЕРНЫХ ИССЛЕДОВАНИЙ
.. 29. 11. 83 г.

Дубна. 1983

Объединенный институт
ядерных исследований
БИБЛИОТЕКА

1. Введение

В настоящее время имеется возможность работы терминальных пользователей CDC-6500 с редактором текстов EDNA /1,2/. Предварительная информация об этом редакторе была дана в Информационном бюллетене ЛВТА № 22 (1981 г.).

Данная работа познакомит пользователей CDC-6500 со всеми возможностями редактора EDNA.

EDNA - это символично-ориентированный редактор, требующий определенных усилий для изучения его обширных возможностей. Если сравнивать редактор EDNA с редактором EDITOR, которым пользуются практически почти все терминальные пользователи, необходимо отметить следующие особенности редактора EDNA:

- а) для загрузки его требуется до 20000 слов памяти в восьмеричном исчислении (загрузка редактора EDITOR требует 40000 слов памяти);
- б) в одной строке можно объединить до 80 команд;
- в) есть возможность организовать циклы команд при редактировании;
- г) практически нет ограничений на длину редактируемых файлов (длина строки - до 638 символов);
- д) можно работать с двумя входными и двумя выходными файлами;
- е) редактирование ведется в пределах страничного буфера (т.е. работа идет не по всему длинному файлу одновременно), что обеспечивает приемлемые времена ответов;
- ж) в пакетном режиме можно запускать целые программы редактирования.

Описание всех команд и возможностей редактора EDNA на английском языке имеется на файле EDNA, II=MANUAL(CY=1 и CY=2).

2. Общие сведения

Программа EDNA работает в одном из двух режимов:

- а) режим объявления файлов, когда пользователь сообщает системе, с какими файлами он будет оперировать;

б) режим редактирования, когда пользователь редактирует файлы.

При вызове редактора EDNA инициируется режим объявления файлов, и только после этого пользователь может перейти в режим редактирования или же отказаться от работы с редактором EDNA. После окончания редактирования пользователь может или отказаться от работы с редактором EDNA или же вернуться к режиму объявления файлов.

Отдельные части редактируемого в данный момент текста находятся на так называемой странице - "PAGE".

Размер страничного буфера (по умолчанию) - 200 строк, 100 из которых выделяются для текущей загрузки текста, а остальные 100 строк служат для возможного расширения редактируемого текста. Размер страничного буфера можно увеличить до 1000 строк.

Указателем текущей позиции при редактировании служит "DOT" (на терминале этот указатель изображается символом ><).

Большинство редакторских команд выполняются относительно указателя DOT, который можно сдвигать вперед или назад на определенное число символов (в одной строке), или на определенное число строк (на одной странице). После того, как одна страница (PAGE) текста отредактирована, а следующая часть текста для редактирования уже передвинулась на PAGE, повторный доступ к ранее отредактированной странице осуществляется только через выход в режим объявления файлов, где этот отредактированный текст задается как новый входной файл.

Кроме указателя DOT редактор EDNA оперирует еще с одним указателем - маркером "MARK", который может использоваться совместно с DOT для ограничения какой-то области текста при определенных задачах редактирования.

3. Начало и конец сеанса

Инициирование работы редактора EDNA подобно вызову редактора EDITOR.

Предварительно сообщаем, что везде далее в тексте примеров то, что набирает пользователь, будет подчеркнуто сплошной чертой.

Чтобы загрузить редактор EDNA в режиме интеркомовских команд, необходимо на запрос COMMAND - набрать EDNA, т.е.:
COMMAND - EDNA - загрузка редактора EDNA.

В результате загрузки на экране появляется ответ системы и запрос:

VERSION 1.3 - ответ системы;
FILES - - запрос.

Запрос FILES - указывает на то, что EDNA работает в режиме объявления файлов и ждет от пользователя сообщения тех имен файлов (результатирующих (OUTPUT) и исходных (INPUT)), с которыми будет работа во время сеанса. После того, как пользователь правильно объявит файлы, EDNA выходит в режим редактирования, о чем сигнализирует появление на экране символа # - запроса редакторских команд.

После редактирования файла, пользователь может закончить работу, набрав команду QQ (см.п.8). В результате текущий сеанс работы редактора EDNA заканчивается и управление переходит к системе ИНТЕРКОМ (на экране появляется запрос COMMAND-).

Схематично один сеанс работы с редактором EDNA можно представить так:

COMMAND - EDNA	- загрузка редактора EDNA;
VERSION 1.3	- ответ системы;
FILES- <u>OUT</u> <INP	- на запрос FILES- пользователь вводит имена выходного (например, OUT) и входного (например, INP) файлов;
#	- запрос на ввод редакторских команд
.----- # <u>QQ</u>	- сеанс работы редакторских команд EDNA
COMMAND-	- запрос на ввод команд ИНТЕРКОМа.

4. Режим объявления файлов

Имена исходных и выходных (результатирующих) файлов объявляются после запроса:

FILES -

Редактор EDNA допускает задание до двух входных и двух выходных файлов, в зависимости от целей пользователя. Формат задания файлов:

OUTPUT1, OUTPUT2 < INPUT1, INPUT2
(имена выходных файлов) (имена входных файлов)

Примеры

I. Создание нового файла.

В случае, когда не имеется готового входного файла, который должен редактироваться, но необходимо создать выходной файл из информации, поступающей с клавиатуры терминала, задание выглядит так:

FILES - OUTFILE

2. Редактирование существующего файла.

Для того, чтобы отредактировать входной файл с именем INFILE, а выходной результат получить на файле OUTFILE, используется следующее задание:

FILES - OUTFILE < INFILE

3. Новый выходной файл не создается.

Чтобы проверить содержимое существующего локального файла INFILE без создания нового выходного файла, используется следующее задание:

FILES - < INFILE

4. Редактирование двух входных файлов.

Для чтения текста как с главного входного файла INFILE, так и с вторичного входного файла INFILE2, на редактируемую страницу и записи выходного результата на один выходной файл OUTFILE используется следующее задание:

FILES - OUTFILE < INFILE, INFILE2

5. Редактирование 1 входного файла и запись результата на 2 выходных файла.

Чтобы прочитать текст с одного входного файла INFILE и редактировать его так, что результирующий текст будет частично на главном выходном файле OUTFIL1, а частично - на вторичном выходном файле OUTFIL2, используется задание:

FILES - OUTFIL1, OUTFIL2 < INFILE

6. Редактирование двух входных файлов и запись результата на 2 выходных файла.

Чтобы прочитать текст как с главного входного файла INFILE1, так и с вторичного входного файла INFILE2, а отредактированный текст записать частично на главный выходной файл OUTFIL1 и частично - на вторичный выходной файл OUTFIL2, используется задание:

FILES - OUTFIL1, OUTFIL2 < INFILE1, INFILE2

Ограничения

1. Все имена файлов должны начинаться с буквы и состоять только из буквенно-цифровых символов. Только первые семь символов имени файла действительно определяют его имя. Так, например, файлы с именами OUTPUTFILE1 и OUTPUTFILE2 не будут восприниматься как разные файлы.

2. Все имена файлов при их задании должны быть различными!

Неверно:

FILES - ED < ED

FILES - ED < EDA, EDA

FILES - ED, ED1 < ED2, ED

3. Как главный, так и вторичный выходные файлы не должны быть присоединенными перманентными файлами.

4. Главный и вторичный входные файлы могут существовать как:

- а) локальные файлы или
- б) присоединенные перманентные файлы.

5. Все объявленные файлы должны находиться на дисках.

Замечания

- а) Указатели обращения к информации в объявленных файлах устанавливаются в начало перед и после использования файлов (т.е. делается автоматическая "перемотка", REWIND).
- б) После правильного объявления файлов EDNA вводит режим редактирования.
- в) Пользовательская команда %A в режиме задания файлов служит для EDNA выходом в режим команд ИНТЕРКМА.
- г) Если объявленный выходной файл уже существует в сеансе, то EDNA проверяет желание пользователя перезаписать информацию на таком файле вопросом:

OVERWRITE EXISTING FILE: FILENAME?
(перезаписать существующий файл: имя?)

Если пользователь в ответ не наберет YES , то EDNA возвращается снова к режиму объявления файлов, т.е. к запросу FILES-.

- д) В разделе 9 будут описаны специальные команды, которые могут выполняться в режиме задания файлов.

5. Режим редактирования файлов

После получения правильных сведений о заданных файлах EDNA входит в режим редактирования.

Запрос на команду в этом режиме имеет вид #.

Команды редактирования имеют форму:

nX или nXS ,

где n - необязательный фактор повторения,

x - определяет вид команды,

s - задает текстовый фрагмент, необходимый для некоторых команд.

В зависимости от команды x, фактор повторения n указывает или на:

- 1) число повторов выполнения команды x или
- 2) область текста, по отношению к которой должна выполняться команда x.

Редакторские команды могут быть объединены в одну командную строку:

zxyzz эквивалентно командной строке xxxzz ,

где x, y и z - обозначения команд.

Командная строка имеет следующие ограничения:

- 1) Заканчивается "возвратом каретки" - клавиша RETURN.
- 2) Не должна содержать более 160 символов.
- 3) Не должна содержать более 80 команд.
- 4) Дополнительные пробелы могут появляться между командами, но не внутри компонент одной команды (т.е. нельзя набирать пробелы между n и x, а также между x и s).

5.1. Фактор повторения

Фактор повторения n может иметь одну из следующих форм:

- +nX положительная десятичная целая переменная. Знак необязателен. Указывает, что команда x должна быть повторена n раз для n текстовых единиц сразу же после указателя позиции DOT (на экране этот символ изображается как ×);
- nX отрицательная десятичная целая переменная. Указывает, что команда x должна быть повторена n раз для n текстовых единиц, предшествующих указателю позиции DOT. Команда -x эквивалентна -1x;
- > x положительное неопределенное число повторов. Указывает, что команда x должна быть повторена для всех текстовых единиц после указателя позиции DOT;

- <x отрицательное неопределенное число повторений. Указывает, что команда x должна быть повторена для всех текстовых единиц, предшествующих указателю позиции DOT;
- @x используется для указания, что команда x действует для всех текстовых единиц в области между указателем позиции DOT и маркером MARK.
- =x используется для указания текстового фрагмента с целью редактирования символов в нем.
 Задание = x эквивалентно набору -nx, где n - количество символов, заданных в качестве текстового фрагмента в команде (типа c, v, g, n, en, i или u), которая выполнялась последней перед командой x. Эта форма объясняется далее при описании команд C, D, J.

Если фактор повторения опущен, то предполагается, что он равен 1. Идентификаторы команд, которые не требуют задания текстового фрагмента s :

A, D, F, EF, J, K, L, M, O, EO, P, EP, Q, R, ER, SD, V, W, EW, (,) , /.

Идентификаторы команд, которые требуют задания текстового фрагмента s :

B, G, H, EN, SC, SK, SL, T, U, X, EQ NE.

Команды c и i могут использоваться с заданием текстового фрагмента s и без него.

Текстовый фрагмент s имеет форму "x...x", где " " - любой ограничительный символ, не появляющийся в текстовой последовательности x...x.

Например:

G _VUY2 _L - ограничительный символ -пробел.
 D/ZEFQQ/ - ограничительный символ -/.

5.2. Выполнение команд

Интерпретация команды имеет две фазы.

1. Делается грамматический разбор командной последовательности, чтобы посмотреть, правильна ли она синтаксически.
2. Команда выполняется с интерпретацией текстовых фрагментов.

Правильный грамматический разбор должен быть завершён до того, как начинает выполняться команда. Следовательно, если найдена синтаксическая ошибка в командной строке, пользователь должен быть уверен,

что ни одна из команд этой строки не повлияет на редактируемый текст. Синтаксическая ошибка указывается повторением той части предыдущей командной строки, где встретилась ошибка, затем следует сообщение об ошибке.

Пример:

#	<u>C/XXX/DZJKL</u>	- пользователь набирает командную строку
	C/XXX/DZ	- EDNA повторяет часть читаемой строки
*	ILLEGAL COMMAND CHAR-Z	- диагностика: незаконный командный идентификатор z
#		- EDNA готов к приему следующей команды;

При выполнении командной последовательности могут быть ошибки разного рода.

Например:

#	<u>K@A</u>	- набирает пользователь;
*	MARK IS UNDEFINED	- выдается диагностика: положение маркера MARK не определено. Команда K успешно завершена;
#		- запрос на следующую команду.

5.3. Суммарная таблица редакторских команд

Команда	Фактор повторения	Значение	Употребление
1	2	3	4
<u>Команды ввода - вывода</u>			
F	+,-	FINISH:	закрывает главный OUTPUT-файл так, что весь текст, записанный на этот файл, начиная с введения команды F до любой следующей -F команды, затирается.
EF	+,-	EDIT FINISH:	то же, что и F-команда, но закрывает вторичный OUTPUT-файл.
O	+, -, >, <, @	OUTPUT:	заводит на OUTPUT-файле n строк, начиная со строки, содержащей DOT.

I	2	3	4
EO	+, -, >, <, @	EDIT OUTPUT:	то же, что и O -команда, но для вторичного OUTPUT - файла.
P	+, >	PAGE:	записывает содержимое текущей страницы на главный OUTPUT -файл, затем читает новую страницу с главного входного файла.
EP	+, >	EDIT PAGE:	то же, что и P -команда, но читает с вторичного входного файла.
R	+, >	READ:	читает строки с главного входного файла до заполнения страницы.
ER	+, >	EDIT READ:	то же, что и R -команда, но читает строки с вторичного входного файла.
W	+, >, @	WRITE:	записывает строки на главный OUTPUT -файл, начиная с начала страницы.
EW	+, >, @	EDIT WRITE:	то же, что и W -команда, но записывает строки на вторичный OUTPUT-файл.

Команды, манипулирующие со строками

A	+, -, >, <, @	ADVANCE:	передвигает DOT на n строк.
K	+, -, >, <, @	KILL:	вычеркивает n целых строк со страницы.
L	+, -, >, <, @	LIST:	показывает n целых строк, начиная со строки, содержащей DOT.

Команды, манипулирующие с символами

B'S'		BRIDGE:	передвигает DOT по строке до тех пор, пока не встретит символ, не указанный в
------	--	---------	---

1	2	3	4
			последовательности символов 'S' , или же не достигнет конца строки.
C'S'	+, -, >, <, @, =	CHANGE:	заменяет n символов от положения DOT символьной последовательностью 'S' .
D	+, -, >, <, @, =	DELETE:	вычеркивает n символов, начиная с DOT.
I'S'		INSERT:	вставляет символьную строку 'S' сразу же перед положением DOT.
J	+, -, >, <, @, =	JUMP:	передвигает DOT по строке на n символов.

Команды поиска текста

G'S'	+	GET:	исследует текст страницы в направлении вперед, начиная с DOT, на присутствие в нем n-ой символьной последовательности 'S' .
H'S'	+	HUNT:	передвигается по страницам "вперед", пока не встретится n-ая символьная последовательность 'S' . Весь исследуемый текст предшествующий строке с n-ой последовательностью 'S' , записывается на главный OUTPUT-файл. Если этот файл уже был закрыт, то весь исследуемый текст теряется.
EH'S'	+	EDIT HUNT:	то же, что и команда H , но исследуется вторичный входной файл.

1	2	3	4
EQ'S'		EQUAL:	эта команда заканчивается успешно, если текст, следующий немедленно за DOT, начинается с символьной последовательности 'S'.
NE'S'		NOT EQUAL:	эта команда подобна команде EQ, только текст, следующий за DOT, не должен начинаться с символьной последовательности 'S'.

Команды, приведенные в этой таблице, используются при исследовании текста на страницах. Имеется еще набор команд, используемый для обмена текстов между страничным буфером и вторичным буфером, называемым "near". Эти команды будут списаны в соответствующем разделе (п.6).

5.4. Команды ввода-вывода

Команды, описанные в этом разделе, не производят никаких редакторских действий над текстом. Команды ввода-вывода ответственны только за передвижение (перепись) текста между входными, выходными файлами и страничным буфером - страницей PAGE.

5.4.I. Команда P.

P.Page.

Формат: nP

Позволено n: +n, >

Эта команда дает возможность пользователю переписать n страниц главного входного файла в выходной файл. По команде P содержимое страницы (PAGE) записывается на главный выходной файл, чистится страница, а затем на эту страницу заносится следующий текст с главного входного файла. Процедура эта повторяется n раз. По команде P текст на страницу заносится до тех пор, пока:

- а) страница не заполнится (количество строк на странице задается параметром N, если предварительно была введена команда /PAGESIZE в режиме объявления файлов, или через подразумеваемое значение этого параметра - 100 строк)
- или б) встретится конец входного файла.

Команда P обычно является первой редакторской командой, служащей для чтения текста с главного входного файла на страницу PAGE для дальнейшего редактирования. Указатель позиции DOT находится в верхней левой позиции страницы. Положение MARK при этом не определено.

Редактирование текста входного файла не может производиться, пока не выполнится команда P.

По команде >P весь главный входной файл постранично переписывается на главный выходной файл, в результате - страница PAGE остается пустой, а указатель DOT (><) расположен в начале пустой страницы.

По команде nP на главный выходной файл переписуются n страниц главного входного файла.

Чтобы продвинуть на страницу PAGE n строк с главного входного файла, используется команда n(RW), где n - целое число (см. далее команды R и W).

5.4.2. Команда EP.

EP . Edit Page.

Формат: nEP

Позволено n: +n, >

По этой команде выполняются те же действия, что и по команде P, за исключением того, что текст читается со вторичного входного файла. Содержимое страницы PAGE записывается на главный выходной файл.

5.4.3. Команда F.

F.Finish.

Формат: nF

Позволено n: +, -

По команде F закрывается главный выходной (OUTPUT) файл. В то время, когда OUTPUT - файл закрыт (или необъявлен), запись любых текстов на этот файл является фиктивной.

Последующая команда -F отменяет действие предыдущей команды F так, что текст может опять записываться на главный выходной файл, который ранее был закрыт.

5.4.4. Команда EF.

EF.Edit Finish.

Формат: nEF

Позволено n: +, -

По команде EF выполняются те же действия, что и по команде F, только закрывается вторичный выходной файл. Команда -EF отменяет действие предыдущей команды EF.

5.4.5. Команда O.

O. Output.

Формат: nO

Позволено n: +n, -n, >, <, @

По этой команде записываются n строк страницы, начиная со строки, содержащей указатель DOT, на главный выходной файл. Текст при этом не убирается со страницы, а положения указателей DOT и MARK - не меняются.

По команде $+no$ на главный выходной файл переписуются n строк страницы, начиная со строки, содержащей указатель DOT (она переписывается первой).

По команде $-no$ на главный выходной файл записываются n строк страницы, предшествующих строке с указателем DOT, исключая строку, содержащую DOT.

По команде $>o$ на главный выходной файл записываются все строки страницы PAGE, начиная со строки с указателем DOT. Строка, содержащая указатель DOT, также записывается.

По команде $<o$ на главный выходной файл записываются все строки страницы PAGE, предшествующие строке, содержащей указатель DOT.

По команде $@o$ на главный выходной файл записываются все строки, расположенные между строк с указателями DOT и MARK.

5.4.6. Команда EO.

EO. Edit Output.

Формат: nEO

Позволено n : $+n, -n, >, <, @$

По этой команде выполняются те же действия, что и по команде o , только запись текста идет на вторичный выходной файл.

5.4.7. Команда R.

R.Read.

Формат: nR

Позволено n : $+n, >$

По этой команде строки, прочитанные с главного входного файла, заносятся в нижнюю часть страницы PAGE. Положения указателей DOT и MARK при этом не меняются.

Выполнение команды nR прекращается в следующих случаях:

- а) n строк считались на страницу PAGE;

- б) встретился признак конца на главном входном файле;
- в) размер страницы достиг максимального числа строк *m*, значение этого параметра определяется командой /PAGESIZE (подразумеваемое значение равно 200 строк).

5.4.8. Команда ER.

ER.Edit Read.

Формат: *nER*

Позволено *n*: *+n, >*

По этой команде выполняются те же действия, что и по команде *r*, за исключением того, что текст читается со вторичного входного файла, а не с главного.

5.4.9. Команда W.

W.Write.

Формат: *nW*

Позволено *n*: *+n, >, @*

По этой команде выполняется передача строк со страницы, начиная с первой строки, на главный выходной файл. Записанные строки убираются со страницы PAGE. Если передается текст, содержащий указатель DOT, то DOT в результате выполнения этой команды помещается слева в начале страницы PAGE. Если передаваемый текст содержит указатель MARK, то после выполнения команды положение MARK не определено.

Фактор повторения *+n* используется для передачи *n* строк со страницы.

Фактор повторения *>* используется для передачи целой страницы.

Фактор повторения *@* служит для передачи всех строк, находящихся между началом страницы и указателем MARK, но строка, содержащая MARK, при этом не передается.

5.4.10. Команда EW.

EW. Edit Write.

Формат: nEW

Позволено n: +n, >, @

По этой команде выполняются те же действия, что и по команде w, за исключением того, что текст записывается на вторичный выходной файл вместо главного выходного файла.

5.5. Команды, манипулирующие со строками

Все команды, описанные в этом разделе, оперируют только с текстом, который находится на странице в тот момент, когда выполняется команда. Ни одна из этих команд не относится к входным или выходным файлам, и эти команды действуют только на целые строки. Выполнение каждой из этих команд осуществляется таким образом, как если бы указатель позиции DOT(><) и указатель MARK располагались в начале строки, для которой команда применяется.

5.5.1. Команда A.

A. Advance.

Формат: nA

Позволено n: +n, -n, >, <, @

В результате выполнения этой команды указатель позиции DOT передвигается на определенное число строк, заданное через n. После того, как эта команда выполнена, DOT располагается слева перед первым символом строки, в которой происходит остановка.

При использовании фактора повторения +n указатель позиции DOT передвигается в начало n-ной строки по отношению к той строке, где первоначально (до введения команды A) находился указатель DOT.

Отрицательный фактор повторения -n используется для передвижения указателя позиции DOT назад по странице на n строк; таким образом, после применения команды -A указатель DOT передвинется в начало предыдущей строки.

Фактор повторения > действует так, что указатель позиции DOT передвигается в конец текущей страницы и помещается в начале пустой строки, следующей за последней строкой на странице.

Фактор повторения < действует так, что указатель позиции DOT передвигается в начало страницы и помещается перед первым символом первой строки на странице.

Фактор повторения @ указывает, что команда A действует только в области текста, заключенного между указателями DOT(><) и MARK, т.е. в результате выполнения команды @A указатель >< пере-

двинется в начало строки, содержащей MARK . Эта команда выполняется независимо от того, где располагается указатель позиции DOT : в предшествующей указателю MARK строке, в той же строке или последующей.

Замечание

Указатель DOT , расположенный в начале строки (перед первым символом строки), на экране не высвечивается.

5.5.2. Команда к.

к. kill.

Формат: nK

Позволено n: +, -, >, <, @

Эта команда используется для того, чтобы вычеркнуть (или "убить") определенное число строк из текста. Местоположение строк, которые должны быть вычеркнуты, определяются положением указателя позиции DOT (><) . Количество вычеркиваемых строк задается значением фактора повторения n .

При употреблении фактора повторения +n со страницы вычеркнутся n строк, начиная со строки, содержащей указатель позиции (DOT), и включая эту строку. Указатель позиции в результате будет располагаться слева, в начале первой невычеркнутой строки.

При использовании фактора повторения -n со страницы вычеркиваются n строк, предшествующих строке, содержащей указатель позиции DOT . Строка, содержащая DOT (><) , не вычеркивается, и положение DOT не меняется.

Фактор повторения > применяется, когда необходимо вычеркнуть все строки текста страницы, начиная со строки, содержащей указатель позиции DOT , и кончая последней строкой страницы PAGE .

Фактор повторения < употребляется, когда нужно вычеркнуть из текста страницы PAGE все строки, предшествующие той, которая содержит указатель позиции DOT .

Фактор повторения @ используется тогда, когда нужно вычеркнуть все строки текста страницы, расположенные между строками с указателями DOT и MARK . Относительно того, вычеркивать или нет строку с указателем DOT , этот фактор действует аналогично факторам

+n или -n , в зависимости от того, в какой строке расположен указатель MARK (после строки с указателем DOT или до нее, соответственно).

5.5.3. Команда L.

L. LIST.

Формат: nL

Позволено n : +, -, >, <, @

Эта команда дает возможность просматривать целые строки текста. Количество изображаемых строк определяется значением фактора n ; строки высвечиваются, учитывая положение указателя позиции DOT .

Фактор повторения +n используется, когда необходимо высветить n строк, начиная со строки, содержащей DOT (и включая эту строку).

Фактор повторения -n используется для изображения n строк, предшествующих той, которая содержит указатель позиции DOT . Строка с указателем позиции DOT (><) - не высвечивается.

Фактор повторения > служит для демонстрации всех строк между той, где находится указатель позиции DOT , и концом страницы PAGE . Строка с указателем DOT - высвечивается на экране.

Фактор повторения < используется для выдачи на терминал всех строк страницы, предшествующих той, где находится указатель позиции DOT .

Фактор повторения @ служит для выдачи на терминал всех строк, заключенных между теми строками, где находятся указатели DOT и MARK . Этот фактор повторения действует также, как факторы +n и -n по отношению к изображению строки, содержащей указатель DOT (в зависимости от того, где находится указатель MARK - после строки с указателем DOT или до нее).

Если указатель позиции находится не в начале строки, то, как уже было сказано, его положение обозначается символом >< .

Например:

Если высвечивается на экране терминала

ВАА,ВАА,ВLACK SHEEP
><

то это означает, что указатель позиции DOT(><) находится сразу же после символа "к!"

Если указатель позиции DOT находится в конце строки, то он изображается символом > .

Например:

Если на экране терминала высвечивается:

ВАА,ВАА,ВLACK SHEEP
>

то это означает, что указатель DOT расположен после последнего символа в строке - "р".

5.6. Команды, оперирующие с символами

Все команды, описанные в этом разделе, оперируют только с символами той строки, где находится указатель позиции DOT, за исключением использования фактора повторения @, когда при выполнении команды принимаются в расчет и символы строки с указателем MARK.

5.6.1. Команда c.

C'S'. Change.

Формат : nC'S'

Позволено n: +, -, >, <, @, =

В результате выполнения этой команды n символов, следующие за указателем позиции DOT (предшествующие DOT, если n < 0), заменяются символьной последовательностью 'S'. DOT в результате располагается слева, сразу же за вставленной последовательностью символов.

Пример:

# L	- показать строку с указанием положения DOT
THIS IS THE CURRENT LINE X	- строка показывается
# 7C"LAST"L	- заменить 7 символов после DOT на последовательность символов "LAST"
THIS IS THE LAST LINE X	- показывается получившаяся строка.

Команда nC"ABCDE" эквивалентна выполнению командной строки:
nDI"ABCDE" (см. далее описание команд D и I).

Команда >C'S' применяется в том случае, если весь текст между DOT и концом строки должен быть заменен последовательностью символов 'S'.

Команда <C'S' используется в случае, если весь текст перед DOT (на той же строке) должен быть заменен последовательностью символов 'S'.

Команда @C'S' употребляется для замены всего текста между DOT и MARK на последовательность символов 'S'. DOT и MARK могут находиться на разных строках, т.е. заменяемый текст может включать признак конца строки.

Фактор повторения = используется, если последняя выполняемая команда содержала текстовый фрагмент, и DOT располагается сразу же после этой символьной последовательности. В этом случае команда =C'S' эквивалентна выполнению команды -nC'S' , где n - есть длина последней используемой символьной последовательности.

Пример:

Команда G'ABCD' = C'ZYXWPQRT' (описание команды G в п.5.7.1)
по своему действию эквивалентна команде:

G'ABCD' -4C'ZYXWPQRT'

Фактор повторения $n=4$ во второй команде определен по количеству символов в текстовом фрагменте ABCD . В результате выполнения этих команд текстовая последовательность ABCD заменяется текстом ZYXWPQRT в строке, где указатель DOT находился после символьной последовательности ABCD ; указатель DOT после выполнения будет находиться за символом T.

Если вводятся команды C' или просто C , то это означает, что текстовая последовательность должна посылаться с терминала во время выполнения команды. EDNA запрашивает:

ENTER TEXT -

(Вводите текст)

После такого запроса текст (без ограничителей) может быть введен. Вводимый таким образом текст может состоять из нескольких строк, т.е. содержать признак конца строки (на терминале типа TEKTRONIX ввод каждой символьной строки заканчивается нажатием клавиши исполнения - RETURN). Набор всего текста завершается посылкой символа = . Количество строк, которое может быть введено таким образом, ограничено максимальным размером страницы (M).

Когда страница PAGE подходит к состоянию заполнения, выдается сообщение:

WARNING - PAGE NEARLY FULL

(Предупреждение - страница почти заполнена)

Когда страница PAGE заполнена, выдается сообщение:

INSERT HALTED - PAGE FULL

Добавление строк заканчивается. Последняя строка может быть потеряна. Перед продолжением работы нужно проверить состояние страницы. Для продолжения добавления текста сначала необходимо использовать команды

W или P .

5.6.2 Команда D.

D. Delete.

Формат : nD

Позволено n: +, -, >, <, @, =

По этой команде происходит вычеркивание со строки n символов, следующих сразу же за указателем позиции DOT . Если n - отрицательно, будут вычеркнуты n символов перед указателем DOT со строки. Если n по величине больше, чем количество символов между DOT и концом этой строки, то все символы между DOT и концом этой строки будут вычеркнуты, но соседние строки останутся нетронутыми.

Фактор повторения > используется в случае, когда необходимо вычеркнуть все символы между указателем позиции (>) и концом строки.

Фактор повторения < используется тогда, когда нужно вычеркнуть все символы: от начала строки до указателя DOT.

Фактор повторения @ используется для вычеркивания всех символов между DOT и MARK . Указатели DOT и MARK могут находиться и на разных строках, в этом случае признаки конца строки будут вычеркнуты.

Команда =D эквивалентна по своему действию команде -nD , где n - количество символов в текстовом фрагменте, используемом в качестве параметра в командах типа B, C, G, H, E, I или U , предшествующих введению команды D .

Пример:

# L	- показать строку, содержащую DOT(L)
THIS IS THE CURRENT LINE	- строка высвечивается
# -4DL	- вычеркнуть 4 символа перед DOT(-4D), показать полученную строку (L)
THIS IS CURRENT LINE	- строка показывается
# >DL	- вычеркнуть все символы в строке после указателя DOT(>D) , по- казать полученную строку (L)
THIS IS	- строка показывается

5.6.3. Команда I.

I. Insert.

Формат: I'S'

В результате выполнения этой команды символьная последовательность 'S' вставляется сразу же перед указателем DOT . После того, как текст 'S' вставлен, указатель DOT находится сразу же за символьной последовательностью S .

Пример:

# <u>L</u>	- показать текущую строку
THIS IS > THE LINE	
# I' NOT REALLY' L	- вставить текст и показать полученную строку
THIS IS NOT REALLY THE LINE >	- строка показывается.

Команда I или I' позволяет пользователю вводить символьный текст во время выполнения команды I , когда высветится запрос:

ENTER TEXT -

(Вводите текст -)

Вводимый пользователем текст может содержать признак конца строки (проставляется после нажатия клавиши исполнения, на ТЕКТРОНИХ - это клавиша RETURN). Ввод текстовой последовательности заканчивается набором символа = (подобно набору текста через команду CREATE при работе с редактором EDITOR в Интер-коме); знак = при этом в текст не включается.

Пример:

# <u>LI'</u>	- показать текущую строку (L) , вставить текстовый фрагмент (I')
THIS IS > THE LINE	- строка показывается на экране
ENTER TEXT-	- запрос системы EDNA на команду I'
<u>NOT REALLY=</u>	- текстовый фрагмент, вводимый пользователем с терминала
# <u>L</u>	- показать текущую строку
THIS IS NOT REALLY > THE LINE	- высвечивается полученная строка

Если в результате добавления текста страница PAGE подходит к насыщению (по количеству строк), то действия пользователя в этом случае подобны действиям, описанным в команде C(CHANGE) для аналогичной ситуации.

5.6.4. Команда J.

J. Jump.

Формат: nJ

Позволено n: +, -, >, <, @, =

В результате выполнения этой команды указатель позиции DOT (X) передвигается по строке вперед на n символов, если n - положительно, или назад на n символов, если n - отрицательно.

Команда > J перемещает указатель DOT после последнего символа текущей строки.

Команда < J передвигает DOT в положение перед первым символом текущей строки.

Команда @ J передвигает DOT в текущую позицию маркера MARK, независимо от того, в одной строке находятся эти указатели (DOT и MARK) или в разных.

Команда =J помещает DOT в позицию, предшествующую символьной последовательности 'S', используемой в предыдущей команде B, C, G, H, E, H, I или U.

Пример:

L 5J L

- показать строку с указателем DOT(L),
передвинуть DOT на 5 символов
вперед (5J), показать полученную
строку (L)

THIS IS THE CURRENT LINE
X

- показывается строка с указателем
DOT

THIS IS THE CURRENT LINE
X

- показывается строка после действия
команды 5J.

5.6.5. Команда B.

B.Bridge.

Формат: B'S'

По этой команде указатель позиции DOT передвигается через любые символы в строке, совпадающие с указанными в последовательности символов 'S'. Передвижение указателя DOT останавливается, когда встречается в строке символ, не указанный среди символов последовательности 'S'. Порядок символов в последовательности 'S' может быть любым. При выполнении команды B указатель DOT не передвигается за границу конца строки (>).

Для правильного понимания действия команды в рассмотрим следующий пример.

- | | |
|----------------------------|---|
| # <u>3L</u> | - показать 3 строки страницы, начиная с первой (DOT находится в начале страницы) |
| AAAA
BBBBBB
CCCCXZZZ | - строки высвечиваются на экране терминала |
| # <u>B'A=C'*'L</u> | - указатель DOT передвинуть за символы (B'A') ; заменить символы A звездочкой (=C'*') , показать полученную строку (L) |
| *
> | - высвечивается полученная строка |
| # <u>AL</u> | - передвинуть указатель DOT на следующую строку (A), показать эту строку (L) |
| BBBBBB | - указатель DOT передвинулся в начало этой строки |
| # <u>AL</u> | - передвинуть указатель DOT на следующую строку (A), показать эту строку (L) |
| CCCCXZZZ | - указатель DOT теперь расположен в начале этой строки |
| # <u>B'C'LB'ZX'L</u> | - а) передвинуть указатель DOT за символы C(B'C') , показать результирующую строку (L)
б) передвинуть указатель DOT за символы Z и X(B'ZX') , показать полученную строку (L) |
| CCCCXZZZ
x | - результат выполнения команд а) |
| CCCCXZZZ
> | - результат выполнения команд б) |
| # <u><JL</u> | - передвинуть указатель DOT в начало строки (<J), показать эту строку (L) |
| CCCCXZZZ | - указатель DOT находится в начале этой строки |
| # <u>B'CZ'L</u> | - указатель DOT передвинуть за символы C и Z в строке (B'CZ') , показать полученную строку (L) |
| CCCCXZZZ
x | - в результате указатель DOT передвинулся только за символы C в строке, т.к. после символа C в строке встретился символ X, не указанный в последовательности символов к команде B . |

5.7. Команды поиска текста

Команды, описанные в этом разделе, могут использоваться для поиска текстового фрагмента в строках редактируемого файла.

5.7.1. Команда G.

G'S'. Get.

Формат: nG'S'

Позволяет n: +

Эта команда позволяет пользователю исследовать текст, находящийся в данный момент на странице (PAGE), для нахождения в нем текстового фрагмента 'S'. Текст исследуется только вперед от положения указателя позиции (DOT). Если поиск был успешным, указатель позиции (>) сразу же перемещается после найденного текстового фрагмента. Если поиск был неудачным (т.е. на странице PAGE не было обнаружено текстовой последовательности 'S'), то выдается следующее сообщение:

* GET SEARCH FAILURE

(Поиск - неудачен)

а указатель позиции (DOT) располагается в начале пустой строки в конце страницы (PAGE).

Фактор повторения +n служит для исследования текста на присутствие в нем n-ного текстового фрагмента 'S'. То есть, если фактор n опущен или равен 1, то будет отыскиваться первый текстовый фрагмент 'S'. Если фактор n равен 2, то будет выбрана строка, в которой текстовый фрагмент 'S' присутствует вторично, и т.д.

Пример:

4L

BAА,BAА BLACK SHEEP,
HAVE YOU ANY WOOL?
YES SIR,NO SIR,
THREE BAGS FULL.

- показать 4 строки текста
- на экране высвечиваются 4 строки, указатель DOT находится в начале первой строки

G/AA/L

BAА,BAА BLACK SHEEP,
>

- найти строку с текстом 'AA' (G/AA/), показать ее (L)
- показывается строка, где нашелся текст 'AA', указатель DOT находится сразу же за первым текстовым фрагментом 'AA'

2G/EE/L

- найти строку, где текст 'EE' встречается вторично (2G/EE/), показать эту строку (L)

THREE BAGS FULL.
X

- строка показывается, указатель DOT
находится после символов 'EE'.

Замечание

Если редактируемый текст был создан при использовании программы TIP (описание об этой программе имеется на файле TEXTA95 , ID=MANUAL), то он будет содержать служебные символы переключения регистров (shift - символы), с помощью которых определяется графическая форма представления текста, следующего за такими служебными символами. Эти shift -символы должны включаться в текстовый фрагмент во время поиска его в тексте, содержащем как строчные, так и заглавные буквы.

Например, при исследовании строк текста для нахождения в нем текстового фрагмента "bill", фрагмент не будет обнаружен, если поиск его вести по команде G/bill/, так как на файле фрагмент "bill" представлен в виде &bill, где & и ! - служебные символы переключения регистров.

5.7.2. Команда n.

n's'. Hunt.

Формат: nH'S'

Позволено n: +n

Эта команда позволяет пользователю осуществить поиск n-ной символьной последовательности, начиная с положения указателя DOT, в прямом направлении. Эта команда отличается от команды G(GET) тем, что исследование текста проводится не только в пределах одной страницы, а просматриваются и последующие страницы главного входного файла, если это требуется.

Если поиск текста прошел успешно, то указатель DOT располагается сразу же после найденного текста, и строка, содержащая DOT, становится первой строкой страницы (PAGE). Все строки, предшествующие этой, которая содержит DOT (иными словами, все строки, не удовлетворяющие поиску текста 'S'), добавляются к главному выходному файлу. После этого определенное число строк читается с главного входного файла на страницу PAGE, чтобы на ней было требуемое число строк (см. команду /PAGESIZE, параметр N).

Если поиск текста 'S' был неудачным, то все содержимое страницы PAGE и оставшаяся информация главного входного файла переходят

на главный выходной файл. В результате этого страница PAGE -
пустая, а DOT расположен в начале страницы PAGE .

5.7.3. Команда EH.

EH'S'. Edit Hunt.

Формат: nEH'S'

Позволено n: +n

Выполнение этой команды идет по той же схеме, что и команды
H(Hunt) , за исключением того, что все новые входные тексты чи-
таются не с главного входного файла, а со вторичного входного файла.
Исследованные тексты записываются на главный выходной файл.

5.7.4. Команда EQ.

EQ. Equal.

Формат: EQ'S'

Эта команда успешно выполняется в случае, когда текст, следу-
ющий сразу же за указателем DOT в строке, начинается с символьной
последовательности 'S'. После успешного завершения команды EQ начи-
нает выполняться следующая за ней команда из введенной пользователем
командной строки. Если же текст в строке, следующий сразу же за ука-
зателем DOT , не начинается с текстового фрагмента 'S' , заданного
в команде EQ , то команда терпит неудачу: диагностика при этом не
выдается, указатель DOT остается на прежнем месте, но следующая
за командой EQ команда из введенной пользователем командной стро-
ки при этом не выполняется, а редактор EDNA выходит к запросу оче-
редной командной строки (на экране появляется символ #).

В основном, эта команда применяется совместно с такими коман-
дами, как B(Bridge) и G(Get) , а также в цикловых структурах
(см. п.7), чем можно достигнуть больших результатов, чем используя
эти команды по отдельности. Команда EQ часто употребляется со сле-
дующими специальными символами в текстовом фрагменте, который заклю-
чается в знаки доллара (\$) , являющимися ограничителями:

- ZEOL\$ - маркер конца строки
- ZEOP\$ - маркер конца страницы
- ZEOF\$ - маркер конца файла .

Пример:

< A - передвинуть указатель DOT в начало
страницы

- # > (B'L'L(EQ\$EOL\$ K) A) - повторить действия а), б) и в) для всех строк страницы (> (....)):
- а) передвинуть DOT через пробелы (B' ')
 - б) если встретился маркер конца строки (EQ\$EOL\$) , то убрать эту строку (K)
 - в) передвинуть указатель DOT на следующую строку (A)
- конец цикла.

В результате выполнения этих команд все пустые строки со страницы PAGE - вычеркиваются.

5.7.5. Команда NE.

NE. Not Equal.

Формат: NE'S'

Эта команда выполняется успешно в том случае, когда текст, следующий за указателем DOT , не совпадает с символьным текстом, заданным в аргументе 'S'. Эта команда применяется в тех же случаях, что и команда EQ.

Рассмотрим следующий пример:

- # < A - передвинуть указатель DOT в начало страницы PAGE;
- # > (NE\$EOL\$L'I'L' A) - повторить следующие действия для всех строк страницы:
- а) если следом за DOT не встретился маркер конца страницы \$EOL\$,
 - б) вставить пробел (I'L') и затем
 - в) передвинуть указатель DOT на следующую строку (A)
- конец цикла.

В результате выполнения этой команды все строки на странице PAGE будут сдвинуты вправо на один пробел.

5.8. Конец раздела

Конец раздела, встречающийся на странице, изображается четырьмя символами:

xEOS

Для целей редактирования конец раздела представляет собой одну строку (`␣EOS`) . Поэтому конец раздела можно передвигать, редактировать или пропускать, используя команды редактирования.

Конец раздела может быть включен в добавляемый текст введением строки `␣EOS` . Эти четыре символа, определяющие конец раздела, могут быть также использованы как текстовый фрагмент в командах поиска текста. Конец рекорда (`␣EOR`) , встречающийся в изображаемых по команде `L,A` (редактор `EDITOR`) строках файла, будет высвечиваться как `␣EOS` при просмотре этого файла по команде `> L` (редактор `EDNA`) .

6. HEAP - буфер

EDNA имеет буфер, называемый HEAP, в котором могут быть спасены как тексты, так и команды. До 32 "входов" может храниться на HEAP (под "входом" понимается отдельный кусок информации), при этом каждый вход имеет свое имя, присвоенное пользователем, для идентификации его. При опознании имени входа HEAP учитываются только первые 10 символов.

Команды, описанные в этом разделе, позволяют заводить входы в HEAP, передвигать их из HEAP, выполнять команды из HEAP и переиспользовать текст, посланный на HEAP.

6.1. Команда SC.

SC'S'. Save Characters.

Формат: nSC'S'

Позволено n: +, -, >, <, @, =

При выполнении этой команды копируются n символов вперед от указателя DOT (назад, если $n < 0$) на HEAP. Этому тексту затем приписывается на HEAP имя 'S'.

Если на HEAP уже есть вход, определенный этим же именем 'S', то старое содержимое входа перекрывается нсвоей информацией. Выполнение этой команды не изменяет содержимого PAGE, а также местонахождения указателей DOT и MARK.

6.2. Команда SK.

SK'S'. Save Kill.

Формат: SK'S'

При выполнении этой команды убирается вход с именем 'S' из HEAP.

6.3. Команда SD.

SD.Save Display.

Формат: SD

При выполнении этой команды выдается перечень текущего набора идентификаторов HEAP-входов.

6.4. Команда SL.

SL'S'. Save Lines.

Формат: nSL'S'

Позволено n: +, -, >, <, @

По этой команде копируются целые строки текста со страницы PAGE на HEAP . Скопированные части текста хранятся на HEAP -входе с именем 'S' . Если вход с таким именем уже существует на HEAP , то старое содержимое этого входа затирается новой информацией.

Текст на странице PAGE , а также местоположение указателей DOT и MARK не меняются при выполнении этой команды.

- Команда +nSL'S' - спасает n строк вперед от DOT , включая строку с указателем DOT.
- Команда -nSL'S' - спасает n строк, предшествующих строке с указателем DOT , не включая эту строку.
- Команда > SL'S' - спасает все строки, начиная с той, которая содержит указатель DOT , и кончая последней строкой страницы.
- Команда < SL'S' - спасает все строки страницы PAGE , предшествующие строке, содержащей указатель DOT , не включая эту строку.
- Команда @SL'S' - спасает все строки между указателями DOT и MARK . Если указатель DOT находится перед указателем MARK в тексте, то строка, содержащая DOT , спасается; но если указатель MARK стоит в тексте перед указателем DOT , то строка, содержащая DOT , не спасается.

6.5. Команда U.

U'S'. Unsave.

Формат: U'S'

При выполнении этой команды текст, спасенный на HEAP под именем 'S' , передвигается на страницу PAGE сразу же перед положением указателя DOT . В результате указатель DOT располагается сразу после конца добавленного текста. Вход с именем 'S' не вычеркивается при этом с HEAP и может быть опять переписан на страницу PAGE.

Если на странице PAGE не хватает места для переписи с HEAP -буфера всего содержимого входа 'S' , то переписывается только та часть, которую возможно переписать, и выдается диагностика:

```
UNSAVE HALTED - PAGE FULL
```

(Перепись остановлена - страница заполнена)

Эта диагностика говорит о том, что на странице PAGE не хватает поля, чтобы сделать перепись всей информации с HEAP -входа на PAGE.

Если пользователю необходимо довести перепись HEAP -входа на страницу до конца, то нужно действовать так, как рекомендуется в описании действий при диагностике "INSERT HALTED" (см.п.13.3).

6.6. Команда X.

X'S'. Execute.

Формат: nX'S'

Позволяет n: +n

В результате выполнения этой команды содержимое HEAP -входа с именем 'S' выполнится как одна или более командных строк.

Все строки на HEAP -входе должны быть синтаксически правильными перед тем, как они начнут выполняться. После завершения выполнения команд HEAP -входа управление возвращается к команде, следующей немедленно за X'S' командой.

Для командной строки на HEAP , которая будет выполняться с помощью X'S' команды, допускается иметь внутри себя X -команды, ссылающиеся на другие входы на HEAP . Предел, до которого можно организовывать циклы-вложения X -команд, ограничен той памятью, которая имеется для EDNA . Один уровень цикла может быть организован просто, но для более сложных структур нужно использовать интеркоммовскую EFL команду перед тем, как вызывается EDNA .

Нужно быть осторожными, когда используется фактор повторения более чем +1, для уверенности, что команда в результате закончится правильно, т.к. все ошибки, обнаруженные системой при выполнении одной из командных строк HEAP -входа на каком-то шаге n очередного выполнения команды nX'S' не влияют на выполнение следующих командных строк и шагов команды nX'S' . Пользователю при этом выдается диагностика о встреченной ошибке, а выполнение команды nX'S' (при $n > 1$) продолжается, пока не исчерпаются все шаги по n .

Следует помнить, что выполнение команды X'S' может прекратиться только при обнаружении синтаксической ошибки в командах на HEAP -входе. При этом выполнение текущей командной строки прерывается, и выдается соответствующая диагностика об ошибке.

6.7. Пример использования HEAP -буфера

G'RENT'=SC'A'L - вводит пользователь:
а) ищется строка с текстом 'RENT'(G)
б) вход A на HEAP должен содержать текст
RENT(SC)
в) показ строки с DOT(L)

CURRENT LINE
>JU'A'L - вводит пользователь:
а) DOT передвигается в конец строки (> J)
б) перед указателем DOT записать содержи-
мое входа A (U)
в) показать полученную строку (L)

CURRENT LINERENT
><

В результате выполнения команды 'A' текст, имеющийся на входе 'A'(RENT), вставляется в строке перед указателем DOT (см. полученную строку, п.в)).

7. Цикловая и условная структура

Формат: $n (...)$; $n (.../...)$

Позволено n: +n, >, @ ; +1

Если поместить командную последовательность в скобки и задать перед ними соответствующий фактор повторения n, то команды, заключенные в скобках, будут повторены столько раз, сколько определено в факторе повторения.

Например:

A 10(L I' 'L A) - вводит пользователь

- Результат:
- а) Указатель DOT передвигается на следующую строку (A)
 - б) Строка показывается (L)
 - в) Два пробела вставляются в начале строки (I' ')
 - г) Строка опять показывается (L)

Действия а), б), в) и г) будут повторены 10 раз, так что цикл, подобный описанному, может использоваться для вставки пробелов в первые позиции определенного числа строк.

В случае, если выполнение командной последовательности внутри цикла (...) потерпело неудачу, т.е. какая-то из команд не выполнялась, то происходит выход из цикла и выполнение продолжится с команды, расположенной сразу за закрывающей скобкой.

Например:

#(EQ\$XXX\$K)AL - вводит пользователь

Команда EQ будет не выполнена, если текст, следующий за DOT, не совпадает с текстом xxx. В результате выполнение командной строки будет продолжено с команды A.

Если же текст, следующий за DOT в строке, совпадает с xxx, тогда вычеркивается эта строка (K), затем DOT передвигается на следующую строку редактируемого текста (A) и показывается эта строка (L).

Для организации цикла с неопределенным числом повторений, действующим до конца страницы, используется фактор повторения > .

Например:

> (G'ABCD'=C'PQSS') < AL - вводит пользователь

Результат: во всех строках текста страницы PAGE, начиная от положения указателя DOT, при обнаружении текстового фрагмента ABCD (G) будет произведена замена этого фрагмента на фрагмент PQSS(=C), после чего происходит выход из цикла и выполняются команды <AL, т.е. указатель DOT передвигается в начало страницы (<A), затем показывается первая строка (L).

Фактор повторения @ используется для управления циклом в том случае, когда команды, заключенные в скобки (...), должны выполняться для строк, находящихся между строкой, содержащей указатель DOT и строкой, содержащей указатель MARK. Положение указателя MARK должно быть определено до и на всем протяжении выполнения цикла.

Выполнение такого цикла идет по схеме:

- а) указатель DOT продвигается на 1 строку при каждом проходе цикла;
- б) направление движения - от DOT к MARK.

Для обеспечения выполнения последовательности типа:

если ... тогда ... в противном случае
(if ... then ... else),

может использоваться символ /.

Командную последовательность такой структуры будем называть условной.

Например:

*(EQ'XXXX'=C'YY'/AL) - вводит пользователь

Результат: командная строка такой условной структуры интерпретируется так:

- а) если в строке за указателем DOT следует текст XXXX(EQ), то этот текст заменяется символами YY(=C), затем - выход из цикла (команды A и L при этом не выполняются);
- б) в противном случае (когда за указателем DOT в строке стоят символы, отличные от XXXX) команда EQ терпит неудачу, поэтому команда =C'YY' не выполняется, а начинают выполняться команды A и L, т.е. указатель DOT передвигается на следующую строку и на экране высвечивается эта строка.

При употреблении командной строки условной структуры пользователю необходимо помнить, что если при выполнении какой-то из команд произошла неудача, т.е. команда не выполнилась, то выполнение командной строки продолжится с команды, следующей сразу же за символом /.

Если же символ / встречается при нормальном выполнении команд условной структуры, то после выполнения всех предшествующих слэшу / команд, происходит выход из цикла (.../...) , команды, расположенные после символа /, при этом не выполняются.

Цикловая структура с символом / не может быть использована в конструкции $n(.../...)$, где $n > 1$.

8. Другие возможности

8.1. Команда v.

V. Verify.

Формат: v

В результате выполнения этой команды EDNA выдает на экран ASCII символ "?" и ждет в ответ один из следующих символов:

- Y - выполнение командной строки продолжается со следующей командой
- N - выполнение текущей командной строки прекращается. Если команда находилась внутри цикла или IF-структуры (условной), выполнение командной строки продолжается с первой командой, следующей за закрывающей скобкой или /
- O - остаток текущего командного цикла игнорируется, и выполнение команды продолжается со следующей итерации командного цикла (если она существует)
- Q - остаток текущей командной строки игнорируется.

Любые другие символы (кроме Y, N, O и Q), набранные пользователем на запрос "?", воспринимаются как N, а лишние символы, введенные случайно пользователем за символами Y, N, O и Q в ответной строке, игнорируются, т.е. EDNA реагирует только на первый символ в ответе пользователя.

Пример.

- | | |
|--------------------------|---|
| # > (G'LINE' L V =D L)AL | - вводит пользователь |
| THIS IS LINE ONE | - первая строка с текстом LINE показывается (G и L) |
| ?Y | - запрос "?" выдается на экран по команде V, пользователь вводит Y для указания, что команда =DL должна выполняться для этой строки |
| THIS IS ONE | - показ измененной строки (L) |
| NOW WE HAVE LINE TWO | - вторая строка, выбранная для исследования (G и L) |
| ?O | - для этой строки игнорируется выполнение = D и L команд (O); далее - переход к следующей итерации цикла |

THIS LINE IS LAST
X

- третья строка, попавшая для исследования (G и L)

?N

- выполнение команды внутри цикла прекращается (N), при выходе из цикла выполняются команды A и L

NOW FOLLOWS THE REST

- указатель DOT перемещается в начало следующей строки (A), высвечивается эта строка (L).

8.2. Команда T.

T'S'. Tabulate.

Формат: T'S'

В результате выполнения данной команды задается табулирующий символ, который в дальнейшем используется для ввода текста, и устанавливаются табулирующие позиции.

В качестве ограничителя символьной строки S выбирается тот же символ, что и табулирующий, а сама символьная последовательность состоит из табулирующих позиций, определяемых как целые числа в возрастающем порядке, разделенных запятыми.

Максимальное число табулирующих позиций равно 20, а максимальное значение номера позиции - 637, которое на 1 меньше, чем максимальная длина строки.

Синтаксис символьной строки S проверяется во время синтаксического анализа командной последовательности, и если синтаксическая ошибка обнаружена, то табулирующий символ становится неопределенным. Последующее применение другого варианта команды T отменяет действие предыдущей и задает новые значения табулирующего символа и позиций.

Пример:

T /10,20,30/

- задаются: табулирующий символ - / и табулирующие позиции - 10,20,30

< J

- указатель DOT помещается в начало строки

I

- вводит пользователь (I)

ENTER TEXT -

- запрос на команду I
(вводите текст -)

<u>value1/value2/row/col</u>	- вводится текст перед DOT
<u>=</u>	
<u># -A</u>	- указатель DOT передвигается в начало введенной строки (-A)
<u># L</u>	- показывается введенная строка (L):
VALUE1	VALUE2 - с 10 позиции,
VALUE2	ROW - с 20 позиции,
ROW	COL - с 30 позиции
COL	
#	-- запрос на следующую команду.

8.3. Команда M.

M. Mark.

Формат: M

При выполнении этой команды задается положение маркера MARK в позиции, определяемой указателем DOT.

Положение маркера MARK должно быть определено этой командой до того, как используется фактор повторения @. Положение маркера MARK остается в указанной позиции неизменным при дальнейших передвижениях указателя DOT до тех пор, пока текст, содержащий указатель MARK не будет выброшен; в таком случае положение MARK становится неопределенным.

Положение указателя MARK может быть переопределено в любое время после выполнения другой команды M.

8.4. Команда %.

% . Comment.

Формат: % текст

Все тексты, лежащие между знаком процента (%) и концом командной строки воспринимаются редактором EDNA как комментарии.

8.5. Команда Q.

Q.Quit.

Формат: Q

Если главный выходной (OUTPUT) файл еще не был закрыт, то в результате выполнения этой команды содержимое страницы PAGE и

остаток главного входного (INPUT) файла копируются на главный выходной файл. Главный и вторичный выходные файлы затем закрываются и EDNA выходит в режим объявления файлов (FILES-mode).

8.6. Команда QQ.

QQ. Quick Quit.

Формат: QQ

Команда QQ подобна команде Q и выполняет те же действия с файлами, но вместо выхода в режим задания файлов выполнение QQ - команды дает возможность из режима работы EDNA выйти сразу в командный режим Интеркома, к запросу COMMAND-.

В результате файлы, образованные в сеансе работы редактора EDNA, оказываются в списке локальных файлов. С этими файлами можно продолжать работу через команды Интеркома.

9. Специальные команды режима

объявления файлов

Эти команды имеют форму:

/ключевое слово

Только первая буква каждого ключевого слова имеет смысл, поэтому ввод такой команды, не имеющей параметра, может быть сокращен до ввода двух символов. Команды режима объявления файлов вводятся пользователем после запроса FILES-.

9.1. Команда /BLANKTRIM (/B)

Формат: FILES-/BLANKTRIM
или FILES-/B

Эта команда меняет состояние BLANKTRIM -флага. Начальное состояние этого флага - OFF. Когда BLANKTRIM- флаг в состоянии ON, то все пробелы, расположенные в конце строк файла, теряются при считывании этих строк на страницу PAGE.

Текущее состояние BLANKTRIM -флага может быть найдено использованием команды /FILES (см. п. 9.4).

Если этот флаг - ON, то в результате выдается сообщение:

- - BLANK TRIM ON - -

9.2. Команда /ERROR=ALL (/E=A)

Формат: FILES-/ERROR=ALL
или FILES-/E=A

По этой команде на экран терминала выдаются все сообщения об ошибках, включая такие, присутствие которых не влияет на выполнение последующих команд (recovered errors). Этот режим сообщения об ошибках - подразумеваемый в системе (включается по умолчанию).

9.3. Команда /ERROR=FATAL (/E=F)

Формат: FILES-/ERROR=FATAL
или FILES-/E=F

По этой команде на экране терминала высвечиваются сообщения только о фатальных ошибках, после которых не выполняются последующие

команды, и нет информации об ошибках, не нарушающих выполнение следующих команд (recovered errors).

9.4. Команда /FILES(/F)

Формат: FILES-/FILES .
или FILES-/F

По этой команде выдается список имен всех файлов, которые использованы в текущем сеансе работы с EDNA. Эти имена группируются в соответствии с тем, в каком качестве соответствующие им файлы использовались: это могли быть входные (INPUT) или выходные (OUTPUT) файлы. Имя перманентного файла отмечается звездочкой (*). Кроме этого указывается состояние BLANKTRIM -флага (если этот флаг в положении ON, то выдается об этом сообщение).

Пример.

FILES-/ <u>FILES</u>	- пользователь вводит команду /FILES (можно набрать /F)
--INPUT--	- EDNA информирует о том, что пользователь
INFILE1	в сеансе оперирует с входным файлом
	INFILE1,
--OUTPUT--	- результат работы записывается на выходные
OUT1 OUT2	файлы OUT1 и OUT2,
--BLANK TRIM ON--	- BLANKTRIM флаг - в положении ON
FILES -	- запрос на следующую команду.

9.5. Команда /PAGESIZE=N,M(/P=N,M)

Формат: FILES-PAGESIZE=N,M
или FILES-/P=N,M

где M - целое число, в пределах от 1 до 1000,
а N - целое число, в пределах от 1 до M.

Эта команда используется для задания максимального размера страницы - M строк, а также размера загружаемой страницы - N строк.

Если параметр M опущен, то максимальный размер страницы неизменный (200 строк).

Команды режима редактирования P(PAGE) и H(HUNT) служат для загрузки текста в страничный буфер. Если PAGE -буфер заполнен до

отказа при выполнении этих команд, то для дальнейшего добавления к тексту новых строк нет места. Следовательно, если после выполнения команд PAGE и HUNT страничный буфер заполнен частично, то возможность добавления новых строк в этот буфер определяется параметром N; максимальное количество строк, которые можно дозагрузить в буфер, находится по разности M-N. Подразумеваемые значения этих параметров: M=200, N=100.

Это значит, что PAGE-буфер может содержать 200 строк, но по PAGE-команде текст читается только в 100 строк буфера, так что остальные 100 строк буфера могут быть использованы для добавления текста .

9.6. Команда /STORE(/S,LFN,ID)

Формат: FILES-/STORE,LFN ID
или FILES-/S,LFN, ID

Эта команда служит для каталогизации (делает перманентным) файла LFN с именем LFN и ID=ID . Имя файла LFN должно быть предварительно объявлено в режиме задания файлов.

Эта команда подобна по своему действию команде Интеркома STORE, LFN, ID и выполняет функции, аналогичные команде NOS/BE: CATALOG, LFN, ID=ID, RP=0.

9.7. Команда /UNLOAD(/U)

Формат: FILES-/UNLOAD,FILE1,FILE2,...,FILEN
или FILES-/U,FILE1,FILE2,...,FILEN

По этой команде происходит отказ от файлов с именами FILE1, FILE2,...,FILEN. Эти имена файлов должны быть предварительно заданы в режиме объявления файлов при работе EDNA. Другие интеркомовские локальные файлы неизвестны редактору EDNA, поэтому на них не может быть ссылки в команде /U. Эта команда по своему действию эквивалентна команде NOS/BE:

RETURN,FILE1,FILE2,...,FILEN.

9.8. Команда /QUIT(/Q)

Формат: FILES-/QUIT
или FILES-/Q

По этой команде прекращается работа редактора EDNA и происходит возврат к режиму интеркомовских команд.

После выполнения этой команды на экране терминала высвечивается информация о времени центрального процессора, израсходованного пользователем за сеанс работы редактора EDNA, после чего выдается запрос на ввод команд Интеркома: COMMAND-.

Пример.

#Q	- эта команда вводится пользователем для выхода в режим объявления файлов
FILE WRITTEN : A	- информации пользователю, что выходной результат записался на файл A
FILES-/Q	- в режиме объявления файлов на запрос FILES- пользователь вводит команду /Q для окончания сеанса редактора EDNA
0.453 CP SEC.	- информация о времени центрального процессора (CP) в секундах
COMMAND-	- запрос на ввод команд Интеркома.

10.

Прерывание работы пользователя

а) Если пользователь в режиме объявления файлов при работе с EDNA вводит приказ (%A), то текущий сеанс работы редактора EDNA прекращается, и управление передается системе Интерком, т.е. на экране появляется запрос:

COMMAND-

б) Если пользователь вводит приказ прерывания (%A) во время грамматического разбора части командной строки, то выполнение этой командной последовательности прерывается и выдается следующее сообщение:

*USER ABORT

PRIOR TO COMMAND EXECUTION

(прерывание произошло до того, как команда начала выполняться)

После этого сообщения EDNA выдает на экран терминала символ #, это запрос на ввод новой команды режима редактирования файлов.

в) Если прерывание вводится во время выполнения командной строки, то по возможности выполнение текущей команды заканчивается до того, как управление возвращается к пользователю.

Пример.

#KAL

- Вводится командная строка

%A

- пользователь прерывает выполнение командной строки

*USER ABORT

- EDNA подтверждает прерывание

LAST COMMAND EXECUTED:A

- последняя выполненная команда - A

#

- запрос на ввод следующей команды.

Пользователь может быть уверен, что команды к и а успешно закончены до того, как произошло прерывание. Содержимое страницы при этом представляется в виде, удовлетворяющем дальнейшему редактированию.

Вводом приказа %A выполнение тех команд, которым необходим ввод данных с клавиатуры (I, V), или команд H, EN, P, EP, Q, исследующих длинные файлы, имеющие более тысячи строк, может быть прекращено немедленно.

г) При фатальной ошибке, встреченной в процессе выполнения командной строки, EDNA закрывает выходные (OUTPUT) файлы, но не делает попытки спасти содержимое страницы (PAGE).

В таком случае следующая диагностика высвечивается на экране:

*CONTENTS OF PAGE BUFFER LOST

(Содержимое страничного буфера потеряно)

д) Когда не хватает времени на выполнение команды, содержимое страницы (PAGE) записывается на главный выходной файл и EDNA выходит в режим команд Интеркома.

На экран выдается следующее сообщение:

```
*TIME LIMIT EXCEEDED
```

(Лимит времени превышен)

```
*PAGE WRITTEN TO PRIMARY OUTPUT
```

(Страница записана на главный выходной файл)

II. Пакетный режим

Редактор EDNA дает возможность терминальным пользователям организовать как интерактивную, так и пакетную работу. Для работы в пакетном режиме загрузка редактора EDNA осуществляется следующим образом:

```
COMMAND-EDNA(inp,list) , где
```

inp - имя файла, на котором записан весь сеанс работы пользователя с редактором EDNA (от режима объявления файлов до отказа от работы с редактором EDNA). Файл inp может быть предварительно записан как перманентный файл с перфокарт или с помощью редакторов EDNA или EDITOR, но может быть и просто локальным файлом, созданным через команды CREATE и SAVE(EDITOR) или команды INSERT и QUIT(EDNA). При создании файла inp нужно учитывать все особенности редактора EDNA (см. пример)

list - имя файла, куда записываются все ответы редактора EDNA.

Пример.

Пусть в сеансе работы с Интеркомом пользователь оперирует с перманентным файлом TESTEDN(ID=LVTА). На файле TESTEDN записана, к примеру, следующая информация:

```
AAAA
```

```
BVVVVVV
```

```
CCCCXZZZ
```

Пользователю необходимо вычеркнуть последний символ а в первой строке файла.

Чтобы осуществить такую редакцию файла, используя пакетный режим работы редактора EDNA, пользователь должен создать файл, на котором будет записан весь сеанс работы для редактора EDNA.

Это можно сделать следующим образом (с использованием редактора EDITOR):

COMMAND-EDITOR

..C

100=OUTEDNA < TESTEDN

110=P

120=3L

130=>JL-DL

140=QQ

150==

..S,A,N

..B

COMMAND-EDNA(A,LIST)

0.128 CP SEC.

COMMAND-FILES

--LOCAL FILES--

OUTEDNA LIST &TESTEDN A

В результате на файле LIST должны быть записаны все ответы редактора EDNA на команды сеанса, введенного с файла A.

Распечатка содержимого файла LIST будет выглядеть следующим образом:

VERSION 1.3

FILES-

≡

END OF INPUT : TESTEDN

≡

AAAA

BBBBBB

CCCCXZZ

- вызов редактора EDITOR
- вводится команда CREATE(C) для создания наборного файла
- объявляются имена входного (TESTEDN) и выходного (OUTEDNA) файлов для EDNA
- команды режима редактирования для редактора EDNA
- выход из сеанса работы с редактором EDNA (команда QQ)
- конец создания наборного файла
- на файл A записывается весь сеанс работы для редактора EDNA
- выход из режима работы с редактором EDITOR
- загрузка пакетного режима работы редактора EDNA
- информация о времени центрального процессора (CP), затраченного на работу с редактором EDNA
- пользователь вводит команду Интеркома FILES, чтобы получить перечень файлов, с которыми работает в сеансе.

- версия редактора EDNA-1.3
- запрос на ввод имен исходных и результирующих файлов и запрос команды (≡)
- ответ на команду P
- символ ≡ на распечатке соответствует ASCII символу #, высвечиваемому на экране терминала как запрос на ввод редакторских команд
- ответ на команду 3L (показываются три строки исходного файла TESTEDN)

```

≡ - запрос следующей редакторской
    команды:
AAAA - ответ на команды > JL
  >
AAA - ответ на команды -DL (первая строка
  > исправлена!)
≡ - запрос следующей редакторской команды
FILE WRITTEN : OUTEDNA - ответ на команду QQ.

```

На файле OUTEDNA записан исправленный вариант файла TESTEDN, т.е. первая строка изменена, а вторая и третья строки оставлены без изменения :

```

AAA      | - содержимое результирующего файла OUTEDNA.
BBBBBBB |
CCCCXZZZ |

```

12. Примеры двух сеансов

12.1. Пример I.

Этот пример продемонстрирует пользователю возможность применения редактора EDNA для создания нового файла без редактирования уже существующего исходного файла. Действия пользователя должны быть направлены на получение программы на языке ФОРТРАН с вводом исправлений при обнаружении ошибок в тексте программы и записи результата на файл NEWPROG.

Пользователю необходимо помнить, что операторы ФОРТРАНа начинаются с 7 позиции, поэтому при наборе текста фортрановской программы нужно набирать перед операторами 6 пробелов или же перед набором текста использовать команду T для задания табулирующей позиции.

COMMAND - EDNA	- вызов редактора EDNA
VERSION 1.3	- ответ системы
FILES - NEWPROG	- объявляется имя выходного файла - NEWPROG
# T;7;	- задаются табулирующий символ (;) и табулирующая позиция (7)
#I	- вводится команда I для набора текста программы
ENTER TEXT-	- запрос от EDNA на ввод текста
<u>;PROGRAM VAR(OUTPUT,TAPE6=OUTPUT)</u>	
<u>;CLMNON X(4,3)</u>	- набор с ошибкой (CLMNON)
<u>;REAL Y(6)</u>	
<u>;CALL IOTS(X,12)</u>	- набор с ошибкой (IOTS)
<u>;CALL IOTA(Y,6)</u>	
<u>;WRITE(6,100)X,Y</u>	- неверная метка формата (100)
<u>;WRITE(6,200)X,Y</u>	- набор строки с правильной меткой формата (200)
<u>200;FORMAT(* X= *,12E9.1,5X,* Y= *,6E9.1)</u>	
<u>;STOP</u>	
<u>;END</u>	
=	- ввод текста заканчивается набором символа =
#<A	- указатель DOT возвращается в начало страницы PAGE

#5L

- показать 5 строк вперед от указателя
DOT

```
PROGRAM VAR(OUTPUT,TAPE6=OUTPUT)
COMMON X(4,3)
REAL Y(6)
CALL IOTS(X,12)
CALL IOTA(Y,6)
```

#G'CLMMON' L

- найти строку с текстовым фрагментом
CLMMON и показать ее

```
CLMMON X(4,3)
```

><

#-6D I/COMMON/L

- вычеркнуть 6 символов перед указателем
DOT, и оставить текстовый фрагмент
COMMON и показать строку

```
COMMON X(4,3)
```

><

#2A 14J L

- передвинуть указатель DOT к строке,
где имеется следующая ошибка и помес-
тить затем DOT между символами T и
S в этой строке, показать строку

```
CALL IOTS(X,12)
```

><

#D I'A' L

- вычеркнуть символ S, заменить его
правильным символом A и показать ре-
зультатирующую строку

```
CALL IOTA(X,12)
```

><

#2AK

- передвинуть указатель DOT на 2 строки
вперед и вычеркнуть эту строку (ту, где
неверная метка формата)

#<A>L

- передвинуть указатель DOT в начало
страницы и показать все содержимое
этой страницы

```
PROGRAM VAR(OUTPUT,TAPE6=OUTPUT)
COMMON X(4,3)
REAL Y(6)
CALL IOTA(X,12)
CALL IOTA(Y,6)
WRITE(6,200) X,Y
```

```

200 FORMAT(* X= x,12E9.1,5X,* Y= *,6E9.1)
STOP
END

```

```

#QQ - Выход из режима работы с редактором
EDNA

FILE WRITTEN : NEWPROG - информации о том, что программа пользо-
вателя в результате записана на файле
NEWPROG

0.388 CP SEC. - сеанс работы EDNA потребовал 0.388
секунд работы центрального процессора

COMMAND - - запрос на ввод команд Интеркома.

```

I2.2. Пример 2.

Этот пример проиллюстрирует пользователю возможности редактора EDNA для редактирования ранее созданного файла, длина которого более 100 строк.

Пусть, к примеру, пользователь должен отредактировать перманентный файл ТЕХТ, записанный на дисках общего пользования с ID=LVTA, CY=2, а результат редактирования записать на файл с именем NEWТЕХТ.

```

COMMAND-ATTACH,ТЕХТ,ИД=LVТА,СУ=2 - вызов перманентного файла
ТЕХТ

COMMAND-EDNA - загрузка редактора EDNA

VERSION 1.3 - ответ системы

FILES-NEWТЕХТ<ТЕХТ - задаются имена исходного
(ТЕХТ) и результирующего
(NEWТЕХТ) файлов

#P - текст с файла ТЕХТ считыва-
ется на страницу PAGE
(считываются первые 100 строк
файла)

#5L - показать первые 5 строк
текста, где имеются ошибки

WHEN THE SAMPLE JOB INPUT THROUGH THE CARD READER
THE OPERATING SYSTEM CALLS A PP ROUTINE TO TRANSLATE
THE JOB STATEMENT AND ASSIGN A PRIORITY TO THE JOB.
NEXT THE PPCOPIES
NEXT THE PP COPIES THE JOB THROUGH A BUFFER TO MASS

```

#19J L	- передвинуть в первой строке текста указатель DOT на 19 символов и показать полученную строку
WHEN THE SAMPLE JOB INPUT THROUGH THE CARD READER	
#I' IS'L	- вставить текстовый фрагмент 'IS перед указателем DOT, показать полученную строку
WHEN THE SAMPLE JOB IS INPUT THROUGH THE CARD READER	
#2A L	- передвинуть указатель DOT на 2 строки вперед и показать эту строку
THE JOB STATEMENT AND ASSIGN A PRIORITY TO THE JOB	
#G"MENT"L	- найти текстовый фрагмент MENT, показать строку
THE JOB STATEMENT AND ASSIGN A PRIORITY TO THE JOB	
#I' 'L	- вставить пробел перед указателем DOT, показать результирующую строку
THE JOB STATEMENT AND ASSIGN A PRIORITY TO THE JOB	
#A L	- передвинуть указатель DOT на 1 строку вперед, показать эту строку
NEXT THE PCOPIES	
#K	- убрать эту строку из текста
#P	- считать следующую страницу с файла ТЕХТ
END OF INPUT : ТЕХТ	- информация о том, что файл ТЕХТ считался до конца
#2L	- показать 2 строки текста
AS EACH CONTROL CARD IS EXECUTED IF IS COPIED ONTO THE SYSTEM AND JOB DAYFILES.	
#Q	- выход из режима редактирования EDNA
FILE WRITTEN : NEWTEXT	- информация о том, что отредактированный текст записан на файле NEWTEXT
FILES - /F	- команда /F вводится для получения информации о файлах, с которыми пользователь оперирует в сеансе

--INPUT--

*TEXT

--OUTPUT--

NEWTEXT

FILES - /Q

0.374 CP SEC.

COMMAND-

- выход из режима работы с редактором EDNA

- информация о времени центрального процессора, израсходованного за сеанс работы EDNA

- запрос на ввод команд Интеркома.

13. Диагностика ошибок

13.1. Ошибки при объявлении файлов

<u>Ошибка</u>	<u>Значение / Исправление</u>
* FILE NAME TABLE FULL	- /UNLOAD файлы перед продолжением работы
* FILE NOT FOUND	- попытка выполнить команды /UNLOAD или /STORE для файла, неизвестного EDNA
* FILE NOT DISTINCT	- все объявленные имена файлов должны быть различными
* ILLEGAL KEYWORD	- неверное ключевое слово в команде / ключевое слово
* ILLEGAL DELIMITER	- встретился неверный разделитель (иной, чем '<' или ',') при объявлении файлов
* ILLEGAL DEVICE TYPE	- все объявленные файлы должны находиться на дисках
* ILLEGAL IDENTIFIER	- имя файла или ID начинаются не с буквы
* INPUT FILE NOT FOUND	- объявленный входной файл не является локальным или присоединенным перманентным файлом
* OVERWRITE EXISTING FILE:NAME?	- существующий файл NAME перезаписывается введением на этот вопрос ответа YES. Если же ответ NO, то перезаписи не происходит, команда выбрасывается
* PAGE LOAD TOO LARGE	- размер загружаемой страницы не вписывается в область от 1 до MAXPAGE (параметр N в команде /PAGESIZE)
* PAGE SIZE TOO LARGE	- размер страницы не вписывается в область от 1 до 1000

- * OUTPUT IS PERM FILE - запись информации не разрешается на присоединенный перманентный файл
- * USER ABORT - EDNA выходит в режим команд Интеркома.

13.2. Ошибки в синтаксисе команд

- * COM STRING TOO LONG - максимальная длина командной строки 160 символов
- * EXCEEDED MAX NESTING - число цикловых скобок ограничено 10
- * FILE NOT DECLARED - попытка выполнить H, P или R (EH, EP или ER) команды, когда главный (вторичный) файл не объявлен
- * ILLEGAL COMMAND CHAR - идентификатор команды отличен от A, B, C, D, F, EF, G, H, EH, I, J, K, L, O, EO, P, EP, Q, R, ER, SC, SD, SK, SL, T, U, V, W, EW, X, EQ, NE,), (, /, M
- * ILLEGAL REP FACTOR - неверно задан фактор повторения в команде
- * ILLEGAL TAB ARGUMENT - только числа или запятые могут быть параметром к команде T
- * TAB PT OUT OF RANGE - числа, определяющие табулирующие позиции в команде T, не должны выходить за пределы от 1 до 637
- * TOO MANY COMMANDS - до 80 команд допускается в командной строке
- * TOO MANY TAB POINTS - максимальное число табулирующих позиций в команде T равно 20
- * UNBALANCED (...) - несбалансированы скобки
- * UNDELIMITED STRING - не найден ограничитель, закрывающий текстовый фрагмент в командах B, C, G, H, EH, SC, SK, SL, U, X, EQ или NE

- | | |
|--|---|
| <ul style="list-style-type: none"> * USER ABORT
PRIOR TO COMMAND EXECUTION * ZERO LENGTH STRING * / IN LOOP STRUCTURE | <ul style="list-style-type: none"> - пользователь прервал команду вводом приказа %A во время синтаксического анализа (до начала выполнения) - текстовый параметр в командах B,G,H,EH,SC,SK,SL,U,X,EQ или NE не должен быть пустым - / (слэш) не позволен в цикловой структуре n(...) , если n > 1, но может использоваться в n(..(../.)). |
|--|---|

13.3. Ошибки при выполнении команд

Три категории сообщений об ошибках могут встретиться при выполнении команд:

- а) информативные сообщения (не помечаются знаками * или \$).
- б) сообщения об ошибках, встреченных при выполнении команд в условных структурах, в результате которых не нарушается выполнение следующих шагов цикла (помечаются знаком \$).
Эта диагностика не высвечивается по команде /ERROR=FATAL .
- в) сообщения о фатальных ошибках, которые произошли при выполнении текущей командной строки (помечаются знаком *), высвечиваются по команде /ERROR=FATAL .

- | | |
|--|--|
| <ul style="list-style-type: none"> * CONTENTS OF PAGE BUFFER LOST * DOT NOT WITHIN PAGE END OF INPUT:FILENAME END- OF-RECORD IGNORED | <ul style="list-style-type: none"> - фатальная ошибка, не было попытки спасти содержимое страницы PAGE - выдается при выполнении команд B,C,D или J, когда указатель DOT находится в конце страницы - это информативное сообщение выдается при выполнении команд H,R,R,EH,EP или EP при достижении конца входного файла с именем FILENAME - это информативное сообщение выдается, когда при введении |
|--|--|

- строк текста читаются %EOR или %EOF. Правильный путь - для обозначения конца рекорда или файла вводить признак конца раздела набором строки %EOS
- EOL ADDED TO LAST LINE IN PAGE - это информативное сообщение выдается по команде I или C, когда вводимый текст в конце страницы PAGE не кончается знаком конца строки %EOL%
- * EQUALS IS UNDEFINED - выдается при попытке использовать фактор повторения =, когда последняя выполняемая команда не была одной из следующих: B, C, G, H, EN, I или U
- * GET SEARCH FAILURE - поиск текста окончился неудачей, указатель DOT в результате находится слева в конце страницы PAGE
- * HEAP NAME NOT FOUND - имя входа, заданное пользователем в командах SK, U или X, не обнаружено в HEAP -буфере
- * HEAP OVERFLOW - содержимое страницы PAGE по размеру превышает HEAP -буфер. Можно повторить те же команды, предварительно уменьшив максимальное число строк на странице PAGE (см. команду /PAGESIZE)
- * HUNT SEARCH FAILURE - поиск текста по команде Hunt окончился неудачей, указатель DOT в результате находится слева в конце входного файла
- INSERT HALTED - добавление строк закончено. Последняя строка при этом может быть потеряна. Поэтому перед продолжением добавления строк необходимо проверить со-

стояние страницы PAGE. Для продолжения добавления строк:

а) использовать команду W для записи строк страницы PAGE на выходной файл или

в) использовать команду P для записи всего содержимого страницы PAGE на выходной файл и считывания новой страницы PAGE текста, далее продолжить добавление строк с начала этой страницы.

Этому сообщению предшествует предупреждение:

WARNING - PAGE NEALY FULL

(страница почти заполнена)

* LINE TOO LONG

- максимальная длина строк - 638 символов

* MARK IS UNDEFINED

- попытка сделать ссылку на указатель MARK в то время, как его положение не определено. Чтобы задать положение указателя MARK - используйте команду M

* PAGE ON EMPTY FILE

- попытка выполнить P или ER команды для чтения входного файла, у которого ранее при чтении уже встретился конец файла

READ HALTED - PAGE FULL

- информативное сообщение, выдаваемое по команде P или ER (чтение обрывается - страница заполнена)

* READ ON EMPTY FILE

- попытка выполнить R или ER команды для чтения с входного файла, у которого ранее при чтении уже встретился конец файла

- | | |
|--|---|
| <p>TAB CHARACTER IGNORED</p> | <p>- информативное сообщение, выдается в случае, когда табулирующий символ был введен после конечной, заданной в команде T, табулирующей позиции. Этот символ считается недействительным</p> |
| <p>* TOO MANY HEAP NAMES</p> | <p>- максимально допустимое число входов на HEAP равно 32. Используйте SK -команду для освобождения входов</p> |
| <p>UNSAVE HALTED - PAGE FULL</p> | <p>- не хватает места на странице PAGE, чтобы закончить выполнение команды U(UNSAVE). Для того, чтобы продолжить выполнение команды, используйте рекомендации, данные для диагностики "INSERT HALTED"</p> |
| <p>* USER ABORT
LAST COMMAND EXECUTED :Z</p> | <p>- выполнение командной строки прервано после исполнения команды Z</p> |
| <p>WARNING - PAGE NEARLY FULL</p> | <p>- информативное сообщение, появляющееся во время набора (добавления) текста (см. пояснения к диагностике "INSERT HALTED").</p> |

В заключение автор выражает благодарность В.П.Ширикову, прочитавшему работу в рукописи, за полезные советы и критические замечания; И.И.Шелонцеву за рекомендацию издать это руководство; группе Н.А.Буздавиной за помощь в подготовке руководства к изданию.

14. Литература

1. TEXT EDITING - EDNA (C.Keen, P.Wilson - University of Adelaide, CC, Pub.No 608, July 1980)
2. EDNA - Beginner's Guide
(University of Adelaide, CC, Pub. No 609, June 1979)

СОДЕРЖАНИЕ

	стр.
I. Введение	I
2. Общие сведения	I
3. Начало и конец сеанса	2
4. Режим объявления файлов	3
5. Режим редактирования файлов	6
5.1. Фактор повторения	6
5.2. Выполнение команд	7
5.3. Суммарная таблица редакторских команд	8
5.4. Команды ввода-вывода	12
5.4.1. Команда P	12
5.4.2. Команда EP	13
5.4.3. Команда F	13
5.4.4. Команда EF	13
5.4.5. Команда O	13
5.4.6. Команда EO	14
5.4.7. Команда R	14
5.4.8. Команда ER	15
5.4.9. Команда W	15
5.4.10. Команда EW	16
5.5. Команды, манипулирующие со строками	17
5.5.1. Команда A	17
5.5.2. Команда K	18
5.5.3. Команда L	19
5.6. Команды, оперирующие с символами	21
5.6.1. Команда C	21
5.6.2. Команда D	23
5.6.3. Команда I	24
5.6.4. Команда J	25
5.6.5. Команда B	25
5.7. Команды поиска текста	27
5.7.1. Команда G	27
5.7.2. Команда H	28
5.7.3. Команда EH	29
5.7.4. Команда EQ	29
5.7.5. Команда NE	30

	стр.
5.8. Конец раздела	30
6. HEAR -буфер	32
6.1. Команда SC	32
6.2. Команда SK	32
6.3. Команда SD	32
6.4. Команда SL	33
6.5. Команда U	33
6.6. Команда X	34
6.7. Пример использования HEAR -буфера	35
7. Цикловая и условная структура	36
8. Другие возможности	39
8.1. Команда V	39
8.2. Команда T	40
8.3. Команда M	41
8.4. Команда %	41
8.5. Команда Q	41
8.6. Команда QQ	42
9. Специальные команды режима объявления файлов	43
9.1. Команда /BLANKTRIM	43
9.2. Команда /ERROR=ALL	43
9.3. Команда /ERROR=FATAL	43
9.4. Команда /FILES	44
9.5. Команда /PAGESIZE	44
9.6. Команда /STORE	45
9.7. Команда /UNLOAD	45
9.8. Команда /QUIT	45
10. Прерывание работы пользователя	47
11. Пакетный режим	48
12. Примеры двух сеансов	51
12.1. Пример 1	51
12.2. Пример 2	53
13. Диагностика ошибок	56
13.1. Ошибки при объявлении файлов	56
13.2. Ошибки в синтаксисе команд	57
13.3. Ошибки при выполнении команд	58
14. Литература	62