

**СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА**

P11-86-95

В.В.Кореньков, Г.А.Коробова, С.В.Семашко

РАБОТА

**С УПАКОВАННОЙ ТЕКСТОВОЙ ИНФОРМАЦИЕЙ
НА ЕС ЭВМ**

1986

На базовых ЭВМ ОИЯИ в настоящее время большим дефицитом является память на магнитных дисках для перманентных и временных файлов. Больших объемов дисковой памяти требуют современные банки данных, дополнительные ресурсы нужны при увеличении количества подключаемых к ЭВМ терминалов и т.д.

Для решения этих проблем существует два пути. Первый путь - увеличение количества накопителей на магнитных дисках большой емкости. Второй путь - повышение эффективности использования существующих накопителей.

Для повышения эффективности использования дисковой памяти в ЛВЭ на ЭВМ ЕС-1055 создана система виртуального диска (VDS) ^{/1/}, а в ЛВТА на ЭВМ ЕС-1060 - система регистрации и сопровождения файлов ^{/2/}. Введение этих систем существенно улучшило организацию работы с перманентными файлами пользователей.

Но есть и еще возможности для совершенствования. На ЭВМ ЕС-1060 были проведены исследования всех текстовых наборов данных пользователей. Целью исследования было оценить частоту появления различных символов в текстовых файлах. В результате оказалось, что текстовые файлы более чем на 2/3 состоят из пробелов. В таблице приведена статистика о частоте и вероятности появления всех символов в текстовых наборах пользователей на ЭВМ ЕС-1060.

На основе полученной информации можно более чем в 3 раза уменьшить объем дисковой памяти, занимаемой текстовыми файлами пользователей.

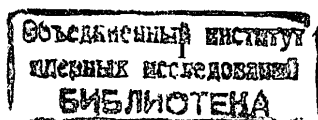
Для реализации этой возможности есть несколько способов упаковки текстовой информации.

1. Код постоянной длины

Один символ на ЕС ЭВМ занимает один байт (8 бит), поэтому цепочку одинаковых символов заменять на один или два байта (хранить признак, количество символов и код символа). Частный случай - цепочку пробелов заменять на один символ, равный количеству пробелов в цепочке (до 64 пробелов).

2. Код переменной длины

Если применять код переменной длины, то эффект от упаковки еще может возрасти. В этом случае длина кода символа (количество бит)



зависит от частоты его появления. Чем чаще появляется символ, тем меньше длина кода (оптимальный код Хаффмана /3/).

В этом случае можно применять дополнительные символы для переключения регистров (кодируя латинские и русские тексты), чтобы уменьшить среднюю длину кода.

Для распаковки текста в этом случае требуется для каждого символа осуществлять поиск по дереву, что приводит к затратам времени центрального процессора. Однако самый большой недостаток кода переменной длины состоит в том, что ошибка в одном бите приводит к неправильной распаковке всего файла. При выборе алгоритма упаковки символов важными факторами являются экономия дисковой памяти, время центрального процессора на упаковку и распаковку информации, надежность хранения текстов. Если исходить из этих факторов, то наибольший эффект получается при использовании кода постоянной длины, что и было взято нами за основу при реализации упаковки текстов.

Другим важным вопросом является определение уровня на котором реализуется выбранный алгоритм упаковки символов.

Можно реализовать упаковку на уровне программ методов доступа. Для этого необходимо изменить системные программы методов доступа, а также программы, реализующие макрокоманды OPEN, CLOSE, FIND, SHOW и другие возможности управления данными. Этот способ труден, но с точки зрения пользователя наиболее предпочтителен, так как распаковка и упаковка происходит "незаметно" для пользователя. Однако при такой реализации есть существенный недостаток: при переходе на новую версию операционной системы нужно много переделывать или делать заново, так как модули управления данными сильно меняются.

Другой путь, который может удовлетворить пользователей, это - создание комплекса каталогизированных процедур для упаковки, распаковки, распечатки, трансляции, копирования текстовых наборов данных, а также работа с упакованными текстами с помощью терминалов. Этот путь был реализован на ЭВМ ЕС-1060 в ЛВТА ОИИИ.

Работа с упакованными текстовыми наборами данных в диалоговой системе TERM

В диалоговой системе TERM /4,5/ различают три типа файлов: наборные, локальные и перманентные. Наборный файл - временный именованный рабочий файл, предназначенный для приема и редактирования различных текстов. Каждому пользователю, работающему за терминалом, отводится один наборный файл, который уничтожается по окончании сеанса работы.

Локальный файл - промежуточный именованный файл пользователя, который служит для спасения наборного файла, для слияния нескольких

файлов, для запуска программы и получения листинга. Количество локальных файлов во время сеанса не превосходит 16.

Перманентный файл - последовательный или библиотечный набор данных длительного хранения.

Диалоговая система предоставляет пользователям возможность переписывать информацию из одного типа файлов в другой.

Команды, реализующие пересылку файлов:

- FETCH - из перманентного в локальный;
- EDIT - из локального в наборный;
- SAVE - из наборного в локальный;
- STORE - из локального в перманентный;
- MERGE - для добавления локального файла в наборный.

Для работы с упакованным форматом текстовой информации в директивах SAVE, EDIT, MERGE существует параметр PACK(P), который сообщает:

а) для команды SAVE - спасти наборный файл в локальный в упакованном виде;

б) для команд EDIT и MERGE - при восстановлении из локального файла в наборный нужно произвести распаковку текстовой информации.

При пересылке между локальным и перманентным файлами никакого преобразования не производится, поэтому упакованные локальные файлы в том же виде переписываются в перманентные и наоборот.

Примеры:

- 1) S ABC P
Спасти наборный файл в локальный с именем ABC в упакованном виде.
- 2) S ABC P M /C/ (1)
Дописать в существующий локальный файл с именем ABC строки из наборного файла, содержащие в первой позиции символ "C". Запись в локальный файл осуществляется в упакованном виде.
- 3) ^E ABC P
Восстановить наборный файл из локального с именем ABC, осуществляя распаковку текстов.
- 4) MER ABC P + 220
Локальный файл с именем ABC добавить к существующему наборному после строки с номером 220 и при добавлении распаковать текст из локального файла.

Программные средства для работы с упакованной текстовой информацией

Для облегчения работы с упакованной текстовой информацией создан набор сервисных программ и каталогизированных процедур для упаковки, распаковки, копирования, трансляции, распечатки текстовых наборов данных.

Все каталогизированные процедуры можно вызывать тремя способами:

- 1) с помощью управляющего оператора EXEC в задании, запущенном в пакетном режиме;
- 2) с помощью команды START с консоли оператора;
- 3) с помощью директив PULF или RUN с терминала, работающего под управлением диалоговой системы TERM.

Пример: распечатку оглавления дискового пакета с именем RES618 можно получить следующими способами:

- 1) // EXEC VTOS, V=RES618 (в пакетном режиме)
- 2) S VTOS. ШИФР V=RES618 (с консоли оператора)
- 3) P S VTOS, V=RES618 (с терминала)

Для работы с упакованной информацией существуют следующие процедуры:

PACK

Эта процедура считывает информацию из входного файла, производит упаковку и полученную информацию записывает в выходной файл. В качестве входного файла может быть последовательный файл на перфокартах, на магнитной ленте, на магнитном диске или раздел библиотечного набора данных на диске. В качестве выходного файла может быть последовательный набор данных или раздел библиотечного набора данных на диске.

Параметры процедуры:

- DSF - имя входного набора данных;
 - DST - имя выходного набора данных;
 - VF - серийный номер тома, на котором расположен набор данных (если он не каталогизирован);
 - VT - серийный номер тома, на котором расположен выходной набор данных (если он не каталогизирован);
- P= { FIX - упаковка кодом фиксированной длины;
NAF - упаковка кодом Хаффмана.

Примеры:

1) // EXEC PACK, DSF='TEXT1', DST='RISKTX'

Каталогизированный последовательный набор данных с именем TEXT1 упаковывается в последовательный набор данных с именем RISKTX.

2) // EXEC PACK, DSF='TEXT1', DST='LIBTX(M1)'

Набор данных с именем TEXT1 упаковывается и записывается в библиотечный набор данных LIBTX (имя раздела M1).

3) // EXEC PACK, DST='LIBTX(MEM2)'

//SYSUT1 DD *

текст на п/к

Набор данных с перфокарт вводится, упаковывается и записывается в библиотечный набор данных LIBTX в качестве раздела с именем MEM2.

PACKALL

Эта процедура считывает все разделы входного библиотечного набора данных, осуществляет упаковку и записывает упакованные разделы в выходной библиотечный набор данных.

Параметры такие же, как в процедуре PACK

Пример:

// EXEC PACKALL, DSF='SYS1.TEACH', DST='TEACH'

Все разделы библиотеки SYS1.TEACH упаковываются и записываются в библиотеку TEACH.

UNPK

Эта процедура выполняет распаковку входного набора данных и записывает в любой последовательный или библиотечный набор данных (обратная процедура PACK).

Пример:

// EXEC UNPK, DSF='TERMIX(MAIN)', DST='TEMP(A)'

Упакованный раздел MAIN библиотеки TERMIX распаковывается и записывается в библиотеку TEMP под именем A.

UNPKALL

Эта процедура считывает все разделы упакованной библиотеки, осуществляет распаковку и записывает в выходной библиотечный набор данных (обратная процедура PACKALL).

Пример:

// EXEC UNPKALL, DSF='TEACH', DSF='ABC', VT=AAAAAA

Разделы упакованной библиотеки TEACH распаковываются и записываются в библиотеку ABC на диске AAAAAA.

UNPKTARE

Эта процедура служит для сброса упакованной библиотеки на магнитную ленту, если эта библиотека переносится на другие ЭВМ, на которых нет средств для работы с упакованной информацией. Для текущего сброса и восстановления упакованных библиотек можно применять процедуры DTMOVE, DTCOPY, TDMOVE, TRCOPY. При работе с этими процедурами никакого преобразования текстов не производится. Для распаковки последовательного файла или раздела библиотеки и записи их на магнитную ленту можно применять процедуру UNPK. Если разделов в библиотечном наборе данных несколько десятков или сотен, то для распаковки их и записи на магнитную ленту можно использовать процедуру UNPKTARE.

Параметры процедуры:

- DSF - имя упакованного библиотечного набора данных;
- VF - серийный номер тома, на котором расположен библиотечный набор данных (Этот параметр задается, если набор данных не каталогизирован);

VT - серийный номер магнитной ленты, на которую записывается информация;

NT - номер файла на магнитной ленте (по умолчанию I);

DST - имя файла на магнитной ленте;

LBT - = $\begin{Bmatrix} SL \\ NL \end{Bmatrix}$ - тип метки;

SL - стандартные метки (по умолчанию);

NL - нет меток.

Пример:

```
// EXEC UNPKTARE, DSF='TERMTX', DSF='FILE', VT=TERMO1
```

Разделы упакованной библиотеки TERMTX распаковываются и записываются первым файлом с именем FILE на магнитную ленту со стандартными метками (серийный номер магнитной ленты TERMO1).

Копирование текстовой информации на магнитную ленту по процедуре UNPKTARE осуществляется одним файлом, причем перед каждым разделом вставляется запись следующего вида:

```
./ ADD NAME = < имя раздела >
```

После этого легко восстановить текстовую библиотеку по утилите IEVUPDTE, задав указанный файл на магнитной ленте входным для данной утилиты ^{16/}.

PR

Эта процедура служит для распечатки текстовой информации как упакованной, так и неупакованной. Текстовая информация может храниться в последовательных или библиотечных наборах данных. Прежде чем распечатать текст, автоматически производится распаковка упакованной информации.

Параметры процедуры:

DS - имя последовательного или библиотечного набора данных, который нужно распечатать;

V - серийный номер тома, где располагается этот набор данных, если он не каталогизирован;

L - список разделов, которые надо распечатать из библиотечного набора данных. Этот список можно задавать различными способами:

L='.ALL' - все разделы (по умолчанию);

L=' префикс . ALL' - разделы, начинающиеся на заданный префикс;

L='M1,M2,...,Mn' - перечисленные имена разделов;

L='M1 - M2' - все разделы от M1 до M2.

Если перечисленные выше списки задать в круглых скобках, то это означает, что эти разделы не надо распечатывать.

Примеры:

```
I) // EXEC PR, DS='SYS1.PROCLIB', L='RDR,TERM,WTR'
```

Распечатать разделы RDR,TERM и WTR из библиотеки SYS1.PROCLIB.

```
2) // EXEC PR, DS='TERMTX', L='A-C(BRDR), MCP.ALL(MCPINIT)'
```

Распечатать все разделы с именами, начинающимися с A до C, исключая раздел BRDR и все разделы, начинающиеся на MCP, исключая MCPINIT.

```
3) // EXEC PR, DS='USERTX', V=NESTOR, L='(A,T.ALL)'
```

Распечатать все разделы библиотеки USERTX (диск NESTOR), кроме раздела A и разделов, начинающихся на T.

COMPC,COMPCL,COMPCLG

Эти процедуры вызывают различные трансляторы, редактор связей и запускают программы на выполнение, причем входной файл для транслятора может быть упакованным.

COMPC - трансляция программ;

COMPCL - трансляция и редактирование связей;

COMPCLG - трансляция, редактирование связей и выполнение.

В этих процедурах много параметров, которые разделяются условно на основные и дополнительные.

Основные параметры описывают входные и выходные файлы, а также режим работы.

DSF - имя последовательного или библиотечного набора данных, где хранятся тексты программ, вызывающихся на трансляцию (могут быть и упакованы);

VF - серийный номер тома этого набора данных, если он не каталогизирован;

L - список разделов, которые вызываются на трансляцию из библиотечного набора данных. Этот список задается аналогично процедуре PR (см. выше);

T - имя транслятора, который вызывается этой процедурой. Возможны варианты:

FG - транслятор FORTRAN-4 уровня G (по умолчанию);

FN - транслятор FORTRAN-4 уровня N;

FX - транслятор FORTRAN N EXTENDED;

FV - транслятор FORTRAN - 77;

AF - ассемблер уровня F;

AN - ассемблер уровня N;

PL - транслятор с языка PL-1;

PA - транслятор с языка PASCAL;

CO - транслятор с языка COBOL.

Для процедуры COMPC основные параметры описаны, остальные параметры касаются в основном процедуры COMPCL, которая обладает широким спектром возможностей.

DST - имя библиотеки загрузочных модулей, в которую записываются оттранслированные программы;

- VT - серийный номер тома библиотеки загрузочных модулей, если она не каталогизирована;
- M - имя раздела в библиотеке загрузочных модулей, под которым пишется готовая к выполнению программа. Если этот параметр задан, то редактор связей объединяет оттранслированные программы в единый загрузочный модуль с указанным именем.

Если параметр m не задан (по умолчанию), то все оттранслированные программы записываются под своими именами в библиотеку загрузочных модулей без реализации внешних ссылок.

Дополнительные параметры этих процедур удовлетворяют стандартным соглашениям.

Все эти параметры - четырехсимвольные, причем первый символ определяет шаг процедуры (с - трансляция, л - редактирование связей, г - выполнение программы). Наиболее употребительны из этих параметров следующие:

- CRGN - запрос на раздел оперативной памяти для трансляции (по умолчанию 200 К);
- CPRM - опции транслятора;
- LLB1 - LLB5 - имена библиотек загрузочных модулей (до 5) для реализации внешних ссылок редактором связей;
- LPRM - опции редактора связей;
- CTIM - заказ времени на выполнение программы;
- GRGN - запрос на раздел оперативной памяти для выполнения программы (по умолчанию 80 К).

Примеры:

1) // EXEC COMPC, DSF='TERMTX', DST='TERMLD', T=AH
Каждый раздел текстовой библиотеки TERMTX распаковывается, обрабатывается ассемблером уровня n и после работы редактора связей записывается в библиотеку загрузочных модулей TERMLD с именем, которое он имел в текстовой библиотеке (без реализации внешних ссылок).

2) // EXEC COMPC, DSF='LIBTX', L='C-F(FDR.ALL)', DST='LIBLD', M=ABC
Разделы текстовой библиотеки LIBTX от с до f, кроме тех, которые начинаются на fdr, обрабатываются транслятором FORTRAN уровня g, объединяются редактором связей в единый загрузочный модуль с именем abc, который записывается в библиотеку LIBLD.

3) // EXEC COMPC, DSF='TEXT', L='A1,A2,RNDM-SIN', LLB2='SYS1.PACKLIB',
// LLB3='SYS1.GENLIB', DST='SYS2.PGMLIB', M='FERMY', T=AH

Из текстовой библиотеки TEXT считаются разделы A1, A2, а также от RNDM до SIN, распаковываются и обрабатываются транслятором FORTRAN уровня n. Затем вызывается редактор связей, который объединяет

оттранслированные модули, используя для реализации внешних ссылок библиотеки SYS1.PACKLIB и SYS1.GENLIB. Полученный загрузочный модуль записывается в библиотеку SYS2.PGMLIB с именем FERMY.

```
4) // EXEC COMPC, DSF='SEMTX', L='TRACE,PUT', T=AH, M=TRACE,
    // DST='SEMLD', GRGN=200K, CTIM=20
```

Из текстовой библиотеки SEMTX считаются разделы TRACE и PUT, распаковываются и обрабатываются ассемблером n. Затем редактор связей объединяет их в один загрузочный модуль с именем TRACE в библиотеке SEMLD. Полученный загрузочный модуль вызывается на выполнение, для чего заказывается 200 К оперативной памяти и 20 минут времени центрального процессора.

Заключение

В результате реализации данной работы стало возможно легко и удобно хранить текстовую информацию (программы, данные, инструкции и т.д.) в упакованном виде на магнитных дисках. При этом в одном текстовом библиотечном наборе данных могут храниться как упакованные, так и неупакованные разделы. Все описанные выше процедуры распаковывают тексты только в том случае, если они были упакованы.

Хранение текстов в упакованном виде позволяет в три раза уменьшить объем дисковой памяти, занимаемой текстовыми наборами данных.

Комплекс созданных каталогизированных процедур позволяет эффективно работать с упакованными текстами как в пакетном режиме, так и с помощью терминалов.

Таблица

Частота употребления символов (только латинские) в текстовых файлах пользователей.

Символ	Код	Число	Частота
Ø	F0	928ØØ6	Ø.11553
1	F1	325347	Ø.Ø4Ø5Ø
E	C5	26Ø726	Ø.Ø3245
T	E3	235644	Ø.Ø2933
,	6B	235417	Ø.Ø293Ø
2	F2	223614	Ø.Ø2783
A	C1	216Ø81	Ø.Ø269Ø
O	D6	21Ø778	Ø.Ø2624
N	D5	2Ø76Ø3	Ø.Ø2584

Таблица (продолжение)

Символ	Код	Число	Частота
*	5C	20443I	∅.∅2545
I	C9	202I57	∅.∅25I6
C	C3	I9433∅	∅.∅24I9
(4D	I78∅88	∅.∅22I7
)	5D	I78∅76	∅.∅22I6
R	D9	I72542	∅.∅2I48
-	6O	I5∅337	∅.∅I87I
3	F3	I48249	∅.∅I845
S	E2	I4795I	∅.∅I84I
L	D3	I39∅35	∅.∅I73∅
M	D4	I33I74	∅.∅I657
P	D7	I27677	∅.∅I589
4	F4	I26485	∅.∅I574
D	C4	I24722	∅.∅I552
.	4B	I244I9	∅.∅I548
5	F5	II37I4	∅.∅I4I5
=	7E	99225	∅.∅I235
6	F6	98725	∅.∅I229
B	C2	933II	∅.∅II6I
8	F8	85597	∅.∅I∅65
H	C8	82382	∅.∅I∅25
7	F7	8I933	∅.∅I∅2∅
F	C6	8I353	∅.∅I∅I2
U	F4	79894	∅.∅∅994
9	F9	78969	∅.∅∅983
/	6I	65994	∅.∅∅82I
X	E7	63669	∅.∅∅792
K	D2	56727	∅.∅∅7∅6
G	C7	56629	∅.∅∅7∅4
+	4E	5344I	∅.∅∅659
,	7D	43∅94	∅.∅∅536
V	E5	36856	∅.∅∅458
Y	E8	359I6	∅.∅∅447
Z	E9	3∅9∅8	∅.∅∅384
I	D1	26∅52	∅.∅∅324
W	E6	2282∅	∅.∅∅284

Таблица (продолжение)

Символ	Код	Число	Частота
Q	D8	I4722	∅.∅∅I83
:	5E	I4722	∅.∅∅I83
:	7A	I∅869	∅.∅∅I35
&	5O	36I∅	∅.∅∅∅44

Общее количество символов: 24I∅536∅.

Из них пробелов: I563I244.

Средняя длина файла: 886∅ символов.

При упаковке кодом постоянной длины: 8493532.

При упаковке кодом Хаффмана: 7647776.

Литература

1. Риднер А., Трофимов В.В., Штрайт Ф., Штрайт Э. ОИЯИ, IO-83-40, Дубна, 1983.
2. Бавижев А.Д., Галактионов В.В., Кореньков В.В. ОИЯИ, II-84-558, Дубна, 1984.
3. Huffman D.A. A Method for the Construction of Minimum-Redundancy Codes, Proc. IRE, 40 No9, 1952.
4. Кореньков В.В. ОИЯИ, PII-82-290, Дубна, 1982.
5. Кореньков В.В., Гончаков В.С. ОИЯИ, BI-II-84-393, Дубна, 1984.
6. Митрофанов В.В., Одинцов Б.В. Программы обслуживания ОС ЕС ЭВМ. "Статистика", М., 1977.

Рукопись поступила в издательский отдел
18 февраля 1986 года.

НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

Вы можете получить по почте перечисленные ниже книги, если они не были заказаны ранее.

D17-81-758	Труды II Международного симпозиума по избранным проблемам статистической механики. Дубна, 1981.	5 р. 40 к.
P18-82-117	Труды IV совещания по использованию новых ядерно-физических методов для решения научно-технических и народнохозяйственных задач. Дубна, 1981.	3 р. 80 к.
D2-82-568	Труды совещания по исследованиям в области релятивистской ядерной физики. Дубна, 1982.	1 р. 75 к.
D9-82-664	Труды совещания по коллективным методам ускорения. Дубна, 1982.	3 р. 30 к.
D3,4-82-704	Труды IV Международной школы по нейтронной физике. Дубна, 1982.	5 р. 00 к.
D11-83-511	Труды совещания по системам и методам аналитических вычислений на ЭВМ и их применению в теоретической физике. Дубна, 1982.	2 р. 50 к.
D7-83-644	Труды Международной школы-семинара по физике тяжелых ионов. Алушта, 1983.	6 р. 55 к.
D2,13-83-689	Труды рабочего совещания по проблемам излучения и детектирования гравитационных волн. Дубна, 1983.	2 р. 00 к.
D13-84-63	Труды XI Международного симпозиума по ядерной электронике. Братислава, Чехословакия, 1983.	4 р. 50 к.
D2-84-366	Труды 7 Международного совещания по проблемам квантовой теории поля. Алушта, 1984.	4 р. 30 к.
D1,2-84-599	Труды VII Международного семинара по проблемам физики высоких энергий. Дубна, 1984.	5 р. 50 к.
D17-84-850	Труды III Международного симпозиума по избранным проблемам статистической механики. Дубна, 1984. /2 тома/	7 р. 75 к.
D10,11-84-818	Труды V Международного совещания по проблемам математического моделирования, программированию и математическим методам решения физических задач. Дубна, 1983	3 р. 50 к.
	Труды IX Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1984 /2 тома/	13 р. 50 к.
D4-85-851	Труды Международной школы по структуре ядра, Алушта, 1985.	3 р. 75 к.

Заказы на упомянутые книги могут быть направлены по адресу:
101000 Москва, Главпочтамт, п/я 79
Издательский отдел Объединенного института ядерных исследований

Кореньков В.В., Коробова Г.А., Семашко С.В. P11-86-95
Работа с упакованной текстовой информацией на ЕС ЭВМ

Работа посвящена актуальной проблеме – повышению эффективности использования дисковой памяти на ЕС ЭВМ. Обосновывается необходимость хранения текстовой информации в упакованном виде. Описывается комплекс программ и процедур для обслуживания текстовых упакованных файлов в пакетном и интерактивном режиме /упаковка, распаковка, распечатка, трансляция, копирование и т.д./.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1986

Перевод О.С.Виноградовой

Korenkov V.V., Korobova G.A., Semashko S.V. P11-86-95
Manipulation with Packed Texts on ES Computer

Actual problem of increasing the efficiency of using disk storage on ES computer is described. Storage of packed texts is provided. A set of subprograms and procedures for manipulation of packed text files in batch and interactive modes is described (packing, unpacking, print, compilation, copying etc).

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1986