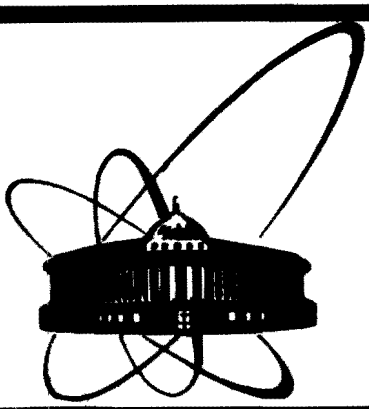


сообщения
объединенного
института
ядерных
исследований
дубна

X 24

P11-87-584

А.М.Хасанов

КОМПЛЕКС ПРОГРАММ ДЛЯ ОБЕСПЕЧЕНИЯ РАБОТЫ С БИБЛИОТЕЧНЫМИ НАБОРАМИ ДАННЫХ ИЗ ЯЗЫКОВ ВЫСОКОГО УРОВНЯ (Pascal, FORTRAN) В ОС ЕС

1987

В операционной системе ОС ЕС широко используются библиотечные наборы данных /I,2/. Они удобны для хранения текстов программ, различных описаний, объектных и загрузочных модулей, других данных. В системе есть специальные программы - утилиты, выполняющие различные операции по обслуживанию библиотечных наборов данных.

Однако не все работы могут быть осуществлены с помощью утилит. Обычно программы, исполняющие более гибкую или специфическую обработку данных в библиотечных наборах, пишутся на ассемблере ЕС ЭВМ с помощью библиотечного метода доступа BPAM /I,2/, что предполагает довольно глубокое знание основ операционной системы.

В работе описывается комплекс программ на ассемблере, дающий возможность работать с библиотечными наборами из распространенных языков Pascal и FORTRAN . По существу этот комплекс реализует библиотечный метод доступа для языков высокого уровня.

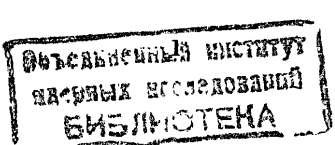
Комплекс предназначен для использования широким кругом программистов при создании собственных программ обработки библиотечных наборов данных. Это могут быть небольшие вспомогательные программы - утилиты, например макросы редактирования. Это могут быть модули, входящие в состав более сложного программного обеспечения типа систем автоматизированного управления. При этом увеличивается производительность труда программиста, поскольку программирование и отладка ведется на языке высокого уровня.

Заметим, что судя по публикации /8/, диалоговая система ISPF/PDF на машинах фирмы IBM обеспечивает прикладного программиста набором сервисных функций, аналогичным описываемому комплексу.

Итак, программы комплекса позволяют:

- открывать для чтения разделов и изменения справочника любую каталогизированную библиотеку (входная библиотека);
- открывать для записи разделов любую каталогизированную библиотеку (выходная библиотека);
- получать параметры используемых библиотек:
  RECFM,LRECL,BLKSIZE ;
- получать из элементов справочника входной библиотеки основное или дополнительное имя раздела, его относительный адрес в библиотеке (TTR);

© Объединенный институт ядерных исследований Дубна, 1987



- изменять справочник входной библиотеки: добавлять, удалять, изменять имена разделов;
- читать содержимое любого раздела входной библиотеки;
- создавать новые разделы в выходной библиотеке, давать им имена;
- копировать разделы из входной библиотеки в выходную;
- копировать входную библиотеку в выходную;
- аннулировать произведенные в выходной раздел записи;
- контролировать результаты описанных выше операций.

С использованием описываемого комплекса были написаны три программы на языке Pascal-8000 : интерактивная система манипулирования разделами и справочником библиотечных наборов данных /4/, программа распечатки справочника библиотеки в специальном виде ADICT (см. приложение Б) и программа выдачи статистики по использованию библиотек программ общего назначения STATIST .

Использование комплекса

Чтобы воспользоваться комплексом, надо подсоединить на шаге редактирования связей библиотеку с программами комплекса, а также включить в задание две DD -карты, соответственно для входных и выходных библиотек:

```
//LIBIN␣␣DD␣DISP=OLD,UNIT=SYSDA,VOL=SER=RESVOL
//LIBOUT␣DD␣DISP=OLD,UNIT=SYSDA,VOL=SER=RESVOL
```

Здесь RESVOL - имя любого постоянно установленного дискового тома. По желанию в DD -карте может быть описана конкретная библиотека:

```
//LIBIN  DD DSN=LIB.TEXT,DISP=SHR
```

Если используются только входные (или только выходные) библиотеки, то можно ограничиться соответствующей DD -картой. Для языка Pascal надо, кроме того, описать вызываемые программы комплекса операторами PROCEDURE и FORTRAN (см. приложение А).

Как видно, имена библиотек не обязательно фиксируются в DD -картах. Эти карты нужны, чтобы система создала необходимые управляющие блоки. В дальнейшем эти блоки динамически изменяются.

Отметим, что для некоторых программ комплекса важен порядок, в котором они вызываются. Например, нельзя читать раздел, пока библиотека не открыта и т.п.

Пример использования комплекса приведен в приложении Г.

AFRLIB, ATOLIB

Открытие входных библиотек осуществляет программа AFRLIB , а выходных - программа ATOLIB . При этом проверяется, каталогизирована ли библиотека, смонтирован ли нужный том и есть ли библиотека в томе. Обращение:

```
CALL AFRLIB(DSN,CODE,RECFM,BLKSIZE,LRECL)
CALL ATOLIB(DSN,CODE,RECFM,BLKSIZE,LRECL)
```

DSN - массив, в котором помещается имя библиотеки, дополненное справа хотя бы одним пробелом. Если первый же символ массива DSN есть пробел, то это означает, что должна быть открыта та библиотека, которая задана в DD -карте. В этом случае имя этой библиотеки возвращается в параметре DSN;

CODE - код завершения программы. Принимает следующие значения: 0, 23, 24, 25, 26, 27, 28, 29, 30;

RECFM - в этом параметре возвращается формат библиотеки в виде текстовой константы длины 8 байт, например 'FB␣␣␣␣␣␣';

BLKSIZE - здесь возвращается длина блока;

LRECL - здесь возвращается длина логической записи.

Описание кодов завершения программ можно найти в приложении В.

Программы AFRLIB и ATOLIB применяют макрокоманду OPEN с параметром TYPE=J , который говорит системе, что при открытии библиотек необходимо использовать модифицированный блок управления файлами задания (JFCB) /1,2/. Этот блок строится в программах комплекса. В нем заполняются только некоторые поля: имя набора данных, серийный номер тома, поле диспозиции и др. Недостающие данные затем будут извлечены из метки библиотеки DSCB1 .

Изменяется также таблица ввода-вывода задачи (TIOT) . В элементы таблицы, соответствующие операторам LIBIN или LIBOUT , заносится адрес соответствующего блока UCB (блока управления устройством, на котором находится диск с библиотекой).

Если работа с библиотекой закончена, можно заказать другую с помощью этих же программ AFRLIB или ATOLIB . Они закроют предыдущую и откроют новую. Если задать имя, начинающееся с пробела, то будет вновь открыта старая библиотека. Если же имя синтаксически неверно, то программы лишь закроют библиотеку.

ACLOSE

Закрыть обе библиотеки можно, обратившись к программе ACLOSE :

```
CALL ACLOSE
```

ANAME

После того, как библиотека объявлена входной с помощью программы AFRLIB , можно работать с ее справочником, читать разделы и т.д.

Например, программа ANAME извлекает информацию из очередного элемента справочника:

```
CALL ANAME(NAME,TYPE,TTR,CODE)
```

NAME – в этот параметр будет записано очередное имя из справочника, выравненное по левой границе и дополненное справа пробелами, если необходимо, до восьми символов;
TYPE – тип имени. Если TYPE=Ø , то это основное имя раздела, если I, то дополнительное;
TTR – относительный адрес раздела;
CODE – коды завершения: Ø, I, 9, I7.

Применяя в цикле программу ANAME , можно прочитать весь справочник библиотеки, пока не будет достигнут код 9.

Заметим, что если имена нужных разделов известны заранее, то не обязательно использовать программу ANAME .

AREN

С помощью программы AREN можно переименовать раздел входной библиотеки:

```
CALL AREN(OLD,NEW,CODE)
```

OLD – старое имя раздела (основное или дополнительное);
NEW – новое имя;
CODE – коды завершения: Ø, I, 4, 6, 7, 8.

ADEL

Программа ADEL удаляет имя из справочника входной библиотеки:

```
CALL ADEL(NAME,CODE)
```

NAME – имя раздела (основное или дополнительное);
CODE – коды завершения: Ø, I, 4, 8.

Заметим, что для удаления раздела библиотеки, надо уничтожить имена, относящиеся к этому разделу: основное и все дополнительные. (Для определения всех имен разделов можно воспользоваться программой ADICT , см. приложение Б) Если раздел имеет только основное имя, то его удаление означает уничтожение раздела.

ACURNT

Для работы программ ADNAME, AREAD и ACOPY необходимо сделать некоторый раздел текущим с помощью программы ACURNT :

```
CALL ACURNT(NAME,CODE)
```

NAME – основное или дополнительное имя раздела;
CODE – коды завершения: Ø, I, 4, 8.

Программа ACURNT использует для установки текущего раздела макрокоманду FIND , которая определяет относительный адрес раздела и помещает его в поле DCBRELAD в блоке управления данными DCB /I,2/. Соответствующий абсолютный адрес заносится в поле DCBFDAD , так что по следующей макрокоманде READ из раздела будет выбрана его первая запись.

ADNAME

Теперь можно работать с текущим разделом, например, дать ему дополнительное имя, вызвав программу ADNAME

```
CALL ADNAME(NAME,CODE)
```

NAME – дополнительное имя;
CODE – коды завершения: Ø, 3, 6, 7, 8.

Относительный адрес для дополнительного имени берется из поля DCBRELAD блока DCB .

AREAD

Для библиотек с RECFM=U или F можно считывать содержимое текущего раздела в память:

```
CALL AREAD(LINE,LEN,CODE)
```

LINE – поле, в которое должны быть записаны данные;
LEN – количество байтов, которое должно быть прочитано. Здесь же возвращается действительное число записанных в LINE байтов;
CODE – коды завершения: Ø, 3, IØ, I5, I8.

Чтение из раздела производится макрокомандой READ , но не обязательно каждому вызову AREAD соответствует вызов макрокоманды, так как данные сначала считываются в некоторый буфер и уже оттуда пересылаются в поле LINE . Поэтому значение LEN не обязано равняться длине логической записи или блока. Оно может иметь практически любое значение. Например, за одно обращение к AREAD можно считать все содержимое раздела.

Если же значение LEN ≤ Ø , то оно полагается равным длине логической записи LRECL для формата F или длине блока BLKSIZE для формата U .

При возвращении из программы параметр LEN может быть меньше заданного, например, если остаток раздела не содержит нужного количества данных. Если раздел исчерпан, то LEN=0 . Чтобы читать раздел сначала, надо вновь обратиться к программе ACURNT .

ACOPY

В выходной библиотеке можно создавать новые разделы и присваивать им имена. Запись в раздел производится с помощью программ ACOPY или AWRITE .

Программа ACOPY копирует текущий раздел входной библиотеки в выходную:

```
CALL ACOPY(CODE)
```

CODE – коды завершения: Ø, 2, 3, I2, I3, I4, I8, I9, 2Ø.

Заметим, что пока новому разделу не присвоено имя с помощью программ ADD или ADDR (см. ниже), он остается открытым для записи. Таким образом, программу ACOPY можно применять для объединения разделов из одной или нескольких входных библиотек.

Приведем допустимые форматы входной и выходной библиотек при использовании программы ACOPY:

F → F ; U → U ; V → V ; F → U ; U → F

Формат V нельзя смешивать с другими форматами, и только для него накладываются ограничения: длина блока выходной библиотеки должна быть не меньше длины блока, а длина записи должна равняться длине записи входной библиотеки. Копирование для формата V происходит поблочно, тем самым создается точная копия текущего раздела.

Форматы F и U разрешается смешивать друг с другом, причем на длину записи и блока ограничений не накладывается. Например, можно слить разделы из входных библиотек с форматами F и U, записав объединенный раздел в выходную библиотеку с форматом U.

Если длина блока входной и выходной библиотек не совпадает, то происходит переблокировка записей. Копирование ведется через два буфера, равных длине блока, соответственно, входной и выходной библиотеки. Информация из текущего раздела считывается в первый буфер. Второй буфер заполняется из первого. Как только второй буфер будет полон, данные из него выводятся в раздел выходной библиотеки. Если по окончании копирования в буфере еще осталось место, то он пока не выводится. Следующее обращение к программе ACOPY (или AWRITE, см. ниже) вызовет заполнение буфера до конца.

Таким образом, все блоки получаются одинаковыми по длине, за исключением, быть может, последнего, который выводится программами ADD или ADDR.

AWRITE

Запись в раздел выходной библиотеки можно производить и программой AWRITE:

```
CALL AWRITE(LINE,LEN,CODE)
```

LINE – содержит данные для записи в раздел;
LEN – длина поля LINE ; LEN > 0 ;
CODE – коды завершения: Ø, 2, I5, I6, I9, 2Ø.

Так же как в программе AREAD, параметр LEN может принимать любые положительные значения.

Сначала данные заносятся в тот же буфер, что и в программе ACOPY, и только если он заполнен, то выводятся в раздел.

Заметим, что запись в один раздел можно выполнять одновременно обеими программами ACOPY и AWRITE, вызывая их сколько угодно раз. Запись всегда производится в конец раздела. Например, можно копировать разделы из одной библиотеки в другую, снабжая их шапками.

ADD, ADDR

Формирование раздела выходной библиотеки заканчивается, когда мы присваиваем ему имя программами ADD или ADDR:

```
CALL ADD(NAME,CODE)
CALL ADDR(NAME,CODE)
```

NAME – имя раздела;
CODE – коды завершения: ADD – Ø, 6, 7, 8, II, I9, 2Ø;
ADDR – Ø, 5, 7, 8, II, I9, 2Ø.

Если имя NAME уже есть в справочнике выходной библиотеки, то программа ADDR заменяет его, то есть теперь это имя указывает на новый раздел. Программа ADD присваивает имя разделу, только если его еще не было в справочнике.

Если получен код Ø для ADD или коды Ø или 5 для ADDR, то это означает, что вновь созданному разделу присвоено основное имя. Последующими обращениями к этим же программам можно присваивать разделу дополнительные имена.

При первом обращении программы ADD и ADDR выводят в раздел, если необходимо, последний укороченный блок из выходного буфера. Для формата F перед этим в буфер дописывается столько пробелов, чтобы длина блока стала кратной длине записи LRECL выходной библиотеки. Затем выдается макрокоманда STOW, которая записывает признак конца раздела (EOF), формирует и записывает элемент раздела в справочник. Затем выходная библиотека закрывается и открывается вновь, чтобы метка библиотеки DSCB1 /1,2/ правильно отражала состояние библиотеки и чтобы избежать неприятностей при зависании системы.

При последующих обращениях программы ADD и ADDR выполняют только действия по записи в справочник элементов с дополнительными именами.

ACONTR

Прежде чем присваивать созданному разделу имя (основное или дополнительное), можно проверить его наличие в справочнике выходной библиотеки:

```
CALL ACONTR(NAME,CODE)
```

NAME – имя раздела;
CODE – коды завершения: 2, 4, 6, 8.

ABORT

Если новому разделу еще не присвоено имя, то можно аннулировать произведенные в него записи:

```
CALL ABORT(CODE)
```

CODE – коды завершения: Ø, 2, II.

AIEBCP

В описываемом комплексе кроме ACOPY есть программа AIEBCP , которая также выполняет копирование разделов. Укажем причины, по которым она была введена.

Возможно такое использование библиотечных наборов данных, при котором большое значение имеет структура разделов и элементов справочника. Элементы справочника могут иметь поле данных с различной информацией, например с указателями на адреса внутри раздела. В библиотеках загрузочных модулей в элементах справочника записаны такие данные, например, как адрес первого блока текста, характеристики загрузочного модуля, адрес точки входа и другие. Модуль также имеет определенную структуру, например, в нем могут быть блоки разной длины, содержащие управляющие словари (ESD и RLD), блоки с текстами (командами и данными – TXT), блоки с управляющей информацией (EOS и EOM) и другие.

Если использовать для копирования таких разделов программу ACOPY, , то получится раздел из блоков одинаковой длины, а элементы справочника, созданные программами ADD и ADDR , не будут содержать поля данных.

Чтобы сохранить сложную структуру разделов и элементов справочника, нужно использовать программу AIEBCP , которая, в свою очередь, вызывает системную утилиту IEBCOPY /3/. Эта утилита не только сохраняет структуру раздела и элементов справочника, но и исправляет указатели TTRN в поле данных пользователя, а также все дополнительные указатели (в списке примечаний) так, что они всегда указывают нужный адрес.

Естественно, при этом должны выполняться все ограничения, которые накладывает утилита IEBCOPY . Например, нельзя копировать из библиотеки формата U в библиотеку формата F , или для формата U длина блока выходной библиотеки должна быть не меньше длины блока входной и другие.

Итак, программу AIEBCP нужно применять, когда важно сохранить структуру раздела и элемента справочника, а программу ACOPY тогда, когда эта структура не важна, но зато нужно избежать ограничений, накладываемых утилитой IEBCOPY . Обращение к программе:

```
CALL AIEBCP(LIST,NUM,CODE)
```

LIST – список имен раздела или разделов, которые должны быть скопированы. Каждое имя занимает 8 байт и, если необходимо, дополняется справа пробелами;
NUM – число имен в списке LIST ;
CODE – коды завершения: Ø, I, 2, I9, 20, 2I, 22.

Данная программа может копировать за одно обращение любое количество разделов, причем одновременно создаются соответствующие элементы в справочнике выходной библиотеки. Если копируется раздел, имеющий дополнительные имена, то все они должны быть заданы в списке LIST .

Если параметр NUM=Ø или NUM < Ø , то происходит полное копирование входной библиотеки в выходную, соответственно с заменой или без замены одинаковых имен. В этом случае содержимое LIST не имеет значения.

Перед вызовом утилиты IEBCOPY программа AIEBCP закрывает выходную библиотеку, а после возвращения из утилиты вновь открывает ее.

Замечание. Если применяется программа AIEBCP , то в задание должны быть добавлены следующие DD -карты:

```
//SYS     DD UNIT=SYSDA,SPACE=(TRK,1)
//LISPOUT DD SYSOUT=A
```

Здесь SYS описывает временный набор данных для передачи IEBCOPY управляющей информации. В набор данных, описанный в LISPOUT, выдаются сообщения утилиты IEBCOPY .

Если в задание добавить DD -карту

```
//S  DD  SYSOUT=A  ,
```

то при ошибках записи в этот набор будет выводиться содержимое блока ECB , первый байт которого содержит код завершения операции ввода/вывода, уточняющий природу ошибки.

Приложение А

ОПИСАНИЕ ПРОГРАММ КОМПЛЕКСА ПРИ ИСПОЛЬЗОВАНИИ В ЯЗЫКЕ PASCAL

```
PROCEDURE AFRLIB(DSN:STR44;VAR CODE:INTEGER;VAR RECFM:ALFA;
                 VAR BLKSIZE,LRECL:INTEGER);FORTRAN;
PROCEDURE ATOLIB(DSN:STR44;VAR CODE:INTEGER;VAR RECFM:ALFA;
                 VAR BLKSIZE,LRECL:INTEGER);FORTRAN;
PROCEDURE ANAME (VAR NAME:ALFA;VAR TYPN,TTR,CODE:INTEGER);FORTRAN;
PROCEDURE ACLOSE;FORTRAN;
PROCEDURE ACONTR(NAME:ALFA;VAR CODE:INTEGER);FORTRAN;
PROCEDURE ACOPY (VAR CODE:INTEGER);FORTRAN;
PROCEDURE ADEL  (NAME:ALFA;VAR CODE:INTEGER);FORTRAN;
PROCEDURE AREN  (OLDNAME,NEWNAME:ALFA;VAR CODE:INTEGER);FORTRAN;
PROCEDURE ADNAME(NAME:ALFA;VAR CODE:INTEGER);FORTRAN;
PROCEDURE ADDR  (NAME:ALFA;VAR CODE:INTEGER);FORTRAN;
PROCEDURE ADD   (NAME:ALFA;VAR CODE:INTEGER);FORTRAN;
PROCEDURE ACURNT(NAME:ALFA;VAR CODE:INTEGER);FORTRAN;
PROCEDURE AREAD (VAR LINE:STRING;VAR LEN,CODE:INTEGER);FORTRAN;
PROCEDURE AWRITE(LINE:STRING;LEN:INTEGER;VAR CODE:INTEGER);FORTRAN;
PROCEDURE AIEBCP(LIST:STRING;NUM:INTEGER;VAR CODE:INTEGER);FORTRAN;
PROCEDURE ABORT (VAR CODE:INTEGER);FORTRAN;
```

Приложение Б

ADICT – программа распечатки справочника библиотечного набора данных

Выдача состоит из двух частей. В первой выдаются основные имена разделов в алфавитном порядке. Для каждого раздела перечисляются также все его дополнительные имена (алиасы).

Во второй части печатаются только дополнительные имена в алфавитном порядке с указанием, к какому разделу относится каждое дополнительное имя. Выдается также количество разделов и дополнительных имен в библиотеке.

Примеры обращения:

```
// EXEC  ADICT,DS='FT77.GENLIB'  или в системе      TERM /7/
P  S ADICT,DS='SYS1.DUBNA',OC=C
```

Программа ADICT1 выдает дополнительно относительные адреса разделов в библиотеке.

Приложение В

Коды завершения программ

0 : успешное выполнение программы
1 : входная библиотека не открыта
2 : выходная библиотека не открыта
3 : текущий раздел не установлен
4 : имени нет в справочнике
5 : имя не найдено и добавлено
6 : имя уже есть
7 : в справочнике нет места
8 : ошибка ввода/вывода при поиске в справочнике
9 : конец справочника
10 : конец раздела
11 : раздел не был записан/скопирован
12 : несовместимые форматы (один из них V)
13 : неравны LRECL (для формата V)
14 : входной BLKSIZE больше выходного (для V)
15 : запись и чтение для формата V не разрешены
16 : длина меньше или равна нулю
17 : ошибка при чтении справочника: блок пропущен
18 : ошибка при чтении раздела
19 : ошибка при записи в раздел
20 : в библиотеке нет места: сделай компрессию
21 : ошибка в IEBCOPY : уровень 4
22 : ошибка в IEBCOPY : уровень 8
23 : синтаксическая ошибка в имени библиотеки
24 : ошибка ввода/вывода при открытии библиотеки
25 : библиотека не каталогизирована
26 : библиотеки нет в томе
27 : том не установлен
28 : DD –карта не задана
29 : не библиотечный набор данных
30 : редкая ошибка

Приложение Г

```
   ПРОГРАММА ВЫДАЕТ НА ПЕЧАТЬ КАРТЫ-КОММЕНТАРИИ ВСЕХ РАЗДЕЛОВ
   БИБЛИОТЕКИ 'NEWLI '

//     EXEC FOR77CLG,LLB2='SYS1.PGMLIB'
       CHARACTER *8 CURNAM,RECFM,LINE *80,DSN *8
C
       DSN='NEWLIB  '
       CALL AFRLIB(DSN,ICODE,RECFM,IBLK,LRECL)
       IF (ICODE.NE.0) THEN
          PRINT *,' ОШИБКА ПРИ ОТКРЫТИИ БИБЛИОТЕКИ. КОД:',ICODE
          STOP
       END IF
       PRINT *,' БИБЛИОТЕКА:',DSN
C
10     CALL ANAME (CURNAM,ITYPE,ITTR,ICODE)
       IF (ICODE.EQ.-1) THEN
          PRINT *,' КОНЕЦ БИБЛИОТЕКИ'
          STOP
       ELSE IF (ICODE.NE.0) THEN
          PRINT *,' ОШИБКА В СПРАВОЧНИКЕ. КОД:',ICODE
          STOP
       END IF
C
       CALL ACURNT(CURNAM,ICODE)
       IF (ICODE.NE.0) THEN
          PRINT *,' ОШИБКА В ACURNT. КОД :',ICODE
          STOP
       END IF
C
       PRINT *,' РАЗДЕЛ:',CURNAM
C
20     N=80
       CALL AREAD (LINE,N,ICODE)
       IF (ICODE.EQ.1) THEN
          GOTO 10
       ELSE IF (ICODE.NE.0) THEN
          PRINT *,' ОШИБКА ПРИ ЧТЕНИИ РАЗДЕЛА:',ICODE
          STOP
       END IF
C
       IF (LINE(1:1).EQ.'*') THEN
          PRINT *,' ',LINE
       END IF
C
       GOTO 2
       END
//G.LIBIN  DD UNIT=SYSDA,DISP=SHR,VOL=SER=SYSRES
//
```


Литература

I. Тимонин В.И. Операционная система ОС ЕС. Основы функционирования. Финансы и статистика, М., 1983.

2. Хусаинов Б.С. Программирование ввода-вывода в ОС ЕС ЭВМ на языке ассемблера. Статистика, М., 1980.

3. Митрофанов В.В., Одинцов Б.В. Программы обслуживания ОС ЕС ЭВМ. Статистика, М., 1977.

4. Хасанов А.М. tooL – система манипулирования разделами и справочником библиотечных наборов данных ОС ЕС. ОИЯИ, Р-11-87-585, Дубна, 1987.

5. Катцан Г. Язык Фортран-77. Мир, М., 1982.

6. Грогоно П. Программирование на языке Паскаль. Мир, М., 1982.

7. Гончаков В.С., Кореньков В.В. и др. ОИЯИ, Р-11-85-172, Дубна, 1985.

8. Joslin P.H. "System productivity facility", IBM System Journal 20, No.4, 388-406(1981).

Рукопись поступила в издательский отдел
24 июля 1987 года.

НЕТ ЛИ ПРОБЕЛОВ В ВАШЕЙ БИБЛИОТЕКЕ?

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

Д3,4-82-704	Труды IV Международной школы по нейтронной физике. Дубна, 1982.	5 р.00 к.
Д7-83-644	Труды Международной школы-семинара по физике тяжелых ионов. Алушта, 1983.	6 р.55 к.
Д2,13-83-689	Труды рабочего совещания по проблемам излучения и детектирования гравитационных волн. Дубна, 1983.	2 р.00 к.
Д13-84-63	Труды XI Международного симпозиума по ядерной электронике, Братислава, Чехословакия, 1983.	4 р.50 к.
Д2-84-366	Труды 7 Международного совещания по проблемам квантовой теории поля. Алушта, 1984.	4 р.30 к.
Д1,2-84-599	Труды VII Международного семинара по проблемам физики высоких энергий. Дубна, 1984.	5 р.50 к.
Д10,11-84-818	Труды V Международного совещания по проблемам математического моделирования,программированию и математическим методам решения физических задач. Дубна, 1983.	3 р.50 к.
Д17-84-850	Труды III Международного симпозиума по избранным проблемам статистической механики. Дубна,1984./2 тома/	7 р.75 к.
Д11-85-791	Труды Международного совещания по аналитическим вычислениям на ЭВМ и их применению в теоретической физике. Дубна, 1985.	4 р.00 к.
Д13-85-793	Труды XII Международного симпозиума по ядерной электронике. Дубна, 1985.	4 р.80 к.
Д4-85-851	Труды Международной школы по структуре ядра. Алушта, 1985.	3 р.75 к.
Д3,4,17-86-747	Труды V Международной школы по нейтронной физике. Алушта, 1986.	4 р.50 к.
	Труды IX Всесоюзного совещания по ускорителям заряженных частиц. Дубна, 1984. /2 тома/	13 р.50 к.
Д1,2-86-668	Труды VIII Международного семинара по проблемам физики высоких энергий. Дубна,1986. /2 тома/	7 р.35 к.

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

Хасанов А.М. P11-87-584

Комплекс программ для обеспечения работы с библиотечными наборами данных из языков высокого уровня (Pascal, FORTRAN) в ОС ЕС

Описывается комплекс программ на ассемблере ЕС ЭВМ, дающий возможность работать с библиотечными наборами данных из распространенных языков Pascal и FORTRAN. По существу он реализует библиотечный метод доступа для языков высокого уровня. Комплекс предназначен для использования широким кругом программистов при создании собственных программ обработки библиотечных наборов данных.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1987

Перевод О.С.Виноградовой

Khasanov A.M. P11-87-584

Program Package for Work with the Partitioned Data Sets in High Level Languages (Pascal, FORTRAN) in OS ES

The described program package is written in Assembler OS ES. It allows to work with the partitioned data sets in wide-spread languages Pascal and FORTRAN. Essentially, it realises the partitioned access method for high level languages. The package is designed for using by a wide circle of programmers for writting their own programs when working with partitioned data sets.

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

Communication of the Joint Institute for Nuclear Research. Dubna 1987