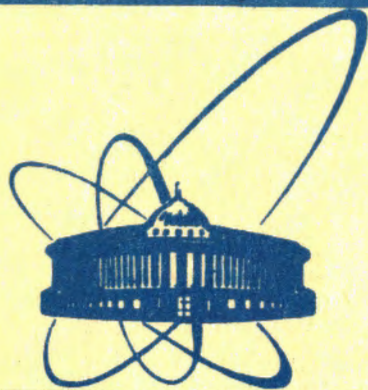


3531/2-80 чр.



сообщения
объединенного
института
ядерных
исследований
Дубна

+

80-251

Б-202

11-80-251

В.К.Балашов, Н.В.Власов, Н.В.Горбунов,
Ю.П.Петухов

СИСТЕМА ВЕДЕНИЯ
ЛИЧНЫХ БИБЛИОТЕК ПРОГРАММ
В РАМКАХ ОПЕРАЦИОННОЙ СИСТЕМЫ "ОС"
НА ЕС ЭВМ

1980

ВВЕДЕНИЕ

Операционная система "ОС" ЕС пакетной обработки и ее язык управления заданиями предоставляют программисту широкий набор системных средств и возможностей^{/1/}, которые, однако, в силу своей универсальности и следования при их создании принципу "все для каждого", чрезвычайно громоздки и неудобны в эксплуатации. Это проявляется при частых отладках больших программ, особенно при работе с наборами данных на магнитных носителях. От программиста требуется хорошее знание языка управления заданиями. Необходимо также большое количество управляющих операторов для системных утилит, трансляторов, редактора связей и т.д.^{/2,3/}.

Использование каталогизированных процедур позволяет упростить процесс подготовки заданий и снижает количество возможных ошибок^{/4,5/}.

В процессе разработки программ и их последующей эксплуатации программисту приходится иметь дело с текстовыми, объектными и загрузочными модулями, которые удобно хранить в отдельных библиотеках в виде их разделов.

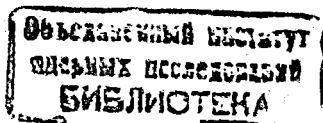
Стандартные каталогизированные процедуры ОС ЕС ориентированы на разовое исполнение небольших программ и не учитывают таких потребностей программиста, как автоматическое занесение в библиотечные наборы данных текстовых и объектных модулей, копирование их и т.д.

Описываемая ниже система ведения личных библиотек по сути есть набор каталогизированных процедур, ориентированных на работу с библиотечными наборами данных /библиотеками/.

Введение символьных параметров в эти процедуры дает в распоряжение программиста "свой", упрощенный вариант языка управления заданиями.

ОРГАНИЗАЦИЯ БИБЛИОТЕК

Обычно программное обеспечение ЭВМ эксплуатируется несколькими группами пользователей, объединенных работой по единой тематике. Каждой такой группе удобно приписать некий индекс и затем с помощью утилиты IENPROGM^{/3/} создать на личном томе группы каталог и логически связать



по индексу верхнего уровня личный том группы с резидентом системы. Это позволяет расширить системный каталог на личный том, т.е. сделать его управляющим.

В каталог управляющего тома библиотеки пользователей заносятся под составными именами в форме:

< Индекс >, < Имя библиотеки > < Тип библиотеки > ,

здесь: < тип библиотеки > = TXIOBILD.

Принято: TX - для библиотеки текстовых модулей,

OB - для библиотеки объектных модулей,

LD - для библиотеки загрузочных модулей.

Процедуры системы ведения личных библиотек находятся в наборе данных SYS1.\$PROCTX, а их сервисные программы - в наборе данных: SYS1.\$SYSTLD на резидентном диске.

Для запуска системы служит модифицированная процедура системного ввода RDR, содержащая в качестве библиотеки процедур сцепленные наборы данных: < индекс >.\$PROCTX и SYS1.PROCLIB, и в качестве дополнительного параметра - IND - индекс пользователя.

Пример: S RDR,00C,IND=SFWR, по умолчанию IND=SYS1.

При использовании общей библиотеки процедур SYS1.\$PROCTX на системном резиденте необходимо при вызове любой процедуры кодировать параметр IND=индекс.

- Пользователь может создать библиотеку процедур < индекс >.\$PROCTX на своем управляющем томе /см. дополнительные возможности, пункт 4.4/. В этом случае индекс пользователя задается лишь один раз при запуске процедуры системного ввода RDR.

Вызов любой процедуры производится по стандартным правилам языка управления заданиями. Все параметры процедур - ключевые; и могут поэтому задаваться в произвольном порядке.

Настоящее описание можно получить с помощью процедуры HELP.

// EXEC HELP

Система ведения личных библиотек обеспечивает:

1. Создание, копирование и уничтожение библиотек.
2. Занесение текстов модулей в библиотеку, редактирование и печать разделов и оглавлений библиотек.
3. Трансляцию, редактирование связей и запуск программ.
4. Дополнительные возможности.

ОПИСАНИЕ БИБЛИОТЕКИ ПРОЦЕДУР

1. Создание, копирование и уничтожение библиотек

1.1. Создание библиотеки (CREATE)

// EXEC CREATE, LIB = имя библиотеки

[, V = архивный номер диска]

[, IND =]

, TX =, OB =, LD =

[, STX = 'SPACE - параметр']

[, SOB = 'SPACE - параметр']

[, SLD = 'SPACE - параметр']

Примечания:

1. Действие процедуры обеспечивается утилитой IENGENDER.
2. Имя библиотеки - идентификатор не должен содержать более шести символов.
3. Параметр V задает место размещения библиотеки. По умолчанию это - управляющий том.
4. Параметрами STX, SOB и SLD задаются размеры создаваемых текстовой, объектной и загрузочной библиотек и их оглавлений. По умолчанию: STX='CYL,(2,1,20)'
SOB='CYL,(1,1,20)'
SLD='CYL,(1,1,10)'
5. Тип и количество создаваемых библиотек определяет набор параметров TX=,OB=,LD=. За один вызов процедуры можно создать таким образом от одной до трех библиотек.

1.2. Копирование "библиотека-библиотека" (COPYLIB)

// EXEC COPYLIB, F = имя исходной библиотеки

, T = имя результирующей библиотеки

, TX=,OB=,LD=

[, M = S|E - способ выборки разделов]

[, L = 'список имен разделов' [SYSIN]

Примечания:

1. Действие процедуры обеспечивается утилитой IEBCOPY.
2. M=S - копируются только разделы из списка L.
=E - разделы из списка L не копируются.
По умолчанию M = S и, если нет списка, копируется вся библиотека.
3. Тип и количество копируемых библиотек определяются набором параметров TX=,OB=,LD=.
4. Копирование производится с заменой разделов с одинаковыми именами.
5. При копировании текстовых и загрузочных библиотек по списку L, для разделов с дополнительными именами необходимо указывать и эти их имена.
6. Процедурой можно сжать библиотеку, для чего достаточно задать одну и ту же библиотеку параметрами F и T.
7. При задании L=SYSIN, список разделов считывается с системного ввода. Имена разделов пробиваются на перфокартах

одно за другим через запятую, начиная с первой позиции, например:

```
// EXEC COPYLIB,F=LIB1,T=LIB2,L=SYSIN  
NAME1,NAME2,...,NAMEX  
/*
```

1.3. Копирование "библиотека-лента" (LIBTAPE)

```
// EXEC LIBTAPE ,LIB= имя библиотеки  
                  ,T = архивный номер ленты  
                  ,TX = ,OB=,LD=  
                  [,N = номер набора]
```

Примечания:

1. Копирование выполняется утилитой IENMOVE на помеченную магнитную ленту.
2. По умолчанию N = 1. Заданные библиотеки копируются в последовательные наборы данных на магнитной ленте.

1.4. Удаление библиотеки (ERASE)

```
// EXEC ERASE       ,LIB = имя библиотеки  
                  ,TX = ,OB=,LD=
```

Примечания:

1. Действие процедуры обеспечивается утилитой IEBGENER.
2. Тип и количество удаляемых библиотек задается набором параметров TX=,OB=,LD=.

1.5. Копирование "лента-библиотека" (TAPELIB)

```
// EXEC TAPELIB ,LIB = имя библиотеки  
                  TX = ,OB=,LD=  
                  ,F = архивный номер ленты  
                  [,T = архивный номер диска ]  
                  [,N = номер набора данных ]  
                  [,STX = 'SPACE-параметр']  
                  [,SOB = 'SPACE-параметр']  
                  [,SLD = 'SPACE-параметр']
```

Примечания:

1. Данная процедура является обратной по отношению к процедурам LIBTAPE и ERASE /см. пункты 1.3 и 1.4/, а именно: на управляющем томе /личном томе, если задан параметр T/

создается библиотека, и с магнитной ленты полностью копируется ее содержимое. По умолчанию N = 1 лента считается стандартно помеченной.

2. Значение параметров процедуры, ответственных за создание библиотек, объясняется в пункте 1.1.

2. Занесение модулей в текстовую библиотеку, редактирование и распечатка разделов и оглавления библиотеки

2.1. Занесение модулей в текстовую библиотеку (TXTPUT)

```
// EXEC TXTPUT ,LIB = имя библиотеки  
                  [,MODE = NEW ]  
                  [, L = 'N= список имен модулей'  
                                                          или  
                                                          = 'I= список индексов модулей']  
                  [, M = S|E - способ выборки модулей]
```

Примечания:

1. Занесение модулей в библиотеку производится утилитой IEBUPDTE. Каждый модуль заносится в отдельный раздел с тем же именем.
2. MODE=NEW - при кодировании этого параметра занесение в библиотеку происходит с заменой модулей с одинаковыми именами.
3. M=S - в библиотеку заносятся только модули из списка L. M=E - модули из списка L в библиотеку не заносятся. По умолчанию M=S и L=ALL, т.е. в библиотеку заносятся все модули. Под индексом понимается идентификация карт в поз. 73-76, например: L='I=J510,D506'.
4. Исходные данные могут быть написаны на языках программирования FORTRAN, ASSEMBLER, PL/1 и в виде макроопределений. Порядок следования модулей - произвольный.
5. Модули, не имеющие имен, заносятся в разделы NOTNAM00, NOTNAM01... . Главному модулю фортрановской программы можно присвоить "имя", поместив в его начало комментарий: с PROGRAM имя программы, тогда он занесется в раздел с этим именем.
6. При вводе исходных модулей с ленты или диска необходимо при вызове процедуры добавить следующую карту:
//MOD.SYSIN DD...

2.2. Редактирование разделов и оглавления библиотеки (EDIT)

```
// EXEC EDIT ,LIB = имя библиотеки  
      [,LIST= ]  
      [,TX = | OB = | LD = ]
```

A/ Редактирование оглавления библиотеки

-] A NAME - создание нового раздела с именем NAME. Раздел вводится с SYSIN.
-] D NAME,NAME1,... - удаление разделов с именами NAME,NAME1,...
-] D .ALL - удаление всех разделов библиотеки.
-] D IND.ALL - удаление всех разделов библиотеки, имена которых начинаются с IND. /IND - идентификатор, от одного до семи символов/.
-] R NAME1,NAME2 - замена имени раздела NAME1 на NAME2.
-] R NAME - замена существующего раздела, новый раздел вводится с SYSIN.

Б/ Редактирование разделов библиотеки

-) U NAME1 [,NAME2] - функциональный оператор, здесь:
NAME1 - имя старого раздела.
NAME2 - имя нового раздела, старый раздел остается без изменения. Этот оператор должен всегда присутствовать во входном потоке, если используются операторы детализации:
-) A N - добавление строк после строки с номером N.
-) R N1 [,N2] - замена строк с номера N1 по N2.
-) D N1 [,N2] - удаление строк с номера N1 по N2.

Примечания:

1. Тип редактируемой библиотеки задается одним из параметров TX=,OB=,LD= по выбору. По умолчанию редактируется текстовая библиотека.
2. Отредактированный текст выводится на печать только при задании параметра LIST =.
3. При занесении раздела с управляющими операторами необходимо при вызове процедуры добавить карту //MOD.SYSIN DD DATA, например:

```
//EXEC EDIT,LIB=PROC  
//MOD.SYSIN DD DATA  
] A BCD  
//BCD PROC  
.....  
/*  
//
```

4. При создании нового раздела или редактировании старого строки нумеруются, начиная с 0 с шагом 1.

2.3. Печать текстовой библиотеки (PRTLIB)

```
// EXEC PRTLIB ,LIB = имя библиотеки  
      [,L = 'список имен разделов' |SYSIN ]  
      [,M = SIE - способ выборки разделов ]
```

Примечания:

1. Разделы библиотеки печатаются в алфавитном порядке.
2. M = S - печатаются только разделы из списка L.
= E - разделы из списка L не печатаются.
По умолчанию M = S и L = ALL, т.е. печатается вся библиотека полностью, значение параметра L=SYSIN /см. пункт 1.2, прим.7/.

```
// EXEC PRTLIB,LIB=HBOOK,L=SYSIN  
HBOOK1,HBOOK2,...NOARG  
FLOARG,...  
/*
```

2.4. Печать оглавления библиотеки (DIRLIB)

```
// EXEC DIRLIB ,LIB = имя библиотеки  
      ,TX =,OB=,LD =
```

Примечания:

1. Действие процедуры обеспечивается утилитой IEHLIST.
2. Тип и количество библиотек определяются набором параметров TX=,OB=,LD=.

3. Трансляция, редактирование связей и запуск программ

3.1. Трансляция (FTN, FTNH,ASM, PLI)

```
// EXEC имя транслятора [,LIB = имя библиотеки [,OB= ]  
      [,R = NO ]  
      [,NAME= имя раздела ]  
      [,COMP= 'режимы трансляции']
```

Примечания:

1. В качестве имени транслятора кодируют FTN, FTNH, ASM и PL1.
2. По умолчанию режимы трансляции - LOAD, NOMAP и NODECK, их можно изменить и дополнить с помощью параметра COMP.
3. При задании параметра OB= объектные тексты модулей обрабатываются редактором связей и заносятся в библиотеку в виде загрузочных модулей с неразрешенными внешними ссылками. В качестве основного и дополнительных имен раздела библиотеки берутся имя подпрограммы и ее ENTRY входы. Главный модуль фортрановской программы /или ассемблерной, не имеющей имени/ заносится в раздел с именем NAME.
По умолчанию NAME=MAIN.
При задании параметра R = NO объектный текст модуля, в библиотеку не заносится, если там уже есть модуль с тем же именем.
4. Для обеспечения работы препроцессора PL1 в качестве библиотеки SYSLIB подключена текстовая библиотека пользователя LIB. Для работы ассемблера в качестве библиотеки SYSLIB подключены сцепленные библиотеки SYS1.MACLIB и LIB. По умолчанию LIB= \$COMN.
5. При вводе исходных модулей с ленты или диска необходимо при вызове процедуры добавить следующую карту:
//MOD.SYSIN DD

3.2. Трансляция и редактирование связей
(FTNLD, FTNHLD, ASMLD, PLILD)

```
// EXEC имя транслятора LD ,LIB = имя библиотеки  
[ ,NAME= имя раздела ]  
[ ,COMP='режимы трансляции' ]  
[ ,EVEN= ]  
[ ,LINK='режимы редактирования' ]  
[ ,LIB1= имя библиотеки для SYSLIB ]  
[ ,LIB2= имя библиотеки для SYSLIB ]  
[ ,LIB3= имя библиотеки для SYSLIB ]
```

Примечания:

1. В качестве имени транслятора кодируют FTN, FTNH, ASM и PL1.

2. По умолчанию режимы трансляции - LOAD, NOMAP и NODECK, их можно изменить и дополнить с помощью параметра COMP.
3. Для разрешения внешних ссылок в качестве библиотеки SYSLIB подключены сцепленные наборы данных LIB, LIB1, LIB2, LIB3 и SYS1.FORTLIB для FTNLD и FTNHLD.
SYS1.PL1LIB для PLILD.
SYS1.MACLIB для ASMLD.
4. Сборка программы невозможна, если в предыдущих шагах задания были зафиксированы коды возврата 8 и выше. Обойти это ограничение можно, определив параметр EVEN=.
5. Стандартно режим редактирования связей NE. Его можно изменить с помощью параметра LINK.
6. Загрузочный модуль программы заносится в раздел с именем NAME. По умолчанию NAME=MAIN, LIB1, LIB2, LIB3= \$COMN.
7. При вводе исходных модулей с ленты или диска необходимо при вызове процедуры добавить следующую карту:
//MOD.SYSIN DD

3.3. Редактирование связей (LNK)

```
// EXEC LNK ,LIB = имя библиотеки  
[ ,NAME= имя раздела ]  
[ ,LINK='режимы редактирования' ]  
[ ,EVEN= ]  
[ ,LIB1= имя библиотеки для SYSLIB ]  
[ ,LIB2= имя библиотеки для SYSLIB ]  
[ ,LIB3= имя библиотеки для SYSLIB ]
```

Примечания:

1. Все параметры имеют тот же смысл, что и в процедурах пункта 3.2.

3.4. Трансляция, редактирование связей и запуск программ
(FTNRUN, FTNHRUN, ASMRUN, PLIRUN)

```
// EXEC имя транслятора RUN [ ,LIB = имя библиотеки ]  
[ ,NAME= имя раздела ]  
[ ,COMP= 'режимы трансляции' ]  
[ ,LINK= 'режимы редактирования' ]  
[ ,LIB1= имя библиотеки для SYSLIB ]  
[ ,LIB2= имя библиотеки для SYSLIB ]  
[ ,LIB3= имя библиотеки для SYSLIB ]  
[ ,PAR = 'параметры программы' ]
```

Примечания:

1. В качестве имени транслятора кодируют FTN, FTNH, ASM и PL1.
2. Все параметры имеют тот же смысл, что и в процедурах пункта 3.2.
Результаты трансляции и редактирования не запоминаются.
3. Доступ к полю параметров PAR можно получить с помощью подпрограммы GOPARM^{7/6}. По умолчанию PAR=пусто.
4. При вводе исходных модулей с ленты или диска необходимо при вызове процедуры добавить следующую карту:
//MOD.SYSIN DD...
5. В шаге запуска программы присутствуют DD операторы языка управления заданиями, обеспечивающие ввод с SYSIN и вывод на печать данных из программы.
При необходимости добавить к шагу запуска программы DD-операторы языка управления заданиями следует кодировать:
//GO.имя DD ...

3.5. Редактирование связей и запуск программ (LNKRUN)

```
// EXEC LNKRUN ,LIB = имя библиотеки
,NAME= имя раздела
[,LINK= 'режимы редактирования']
[,LIB1= имя библиотеки для SYSLIB]
[,LIB2=имя библиотеки для SYSLIB]
[,LIB3= имя библиотеки для SYSLIB]
[,PAR = 'параметры программы']
```

Примечания:

1. Все параметры имеют тот же смысл, что и в процедурах пунктов 3.2. и 3.4.

3.6. Запуск программы (RUN)

```
// EXEC RUN ,LIB = имя библиотеки
,NAME= имя раздела
[,EVEN= ]
[,PAR = 'параметры программы']
```

Примечания:

1. Кодирование параметра EVEN= позволяет запустить модуль, даже если он был помечен редактором связи при его создании как невыполнимый.

2. По умолчанию PAR=пусто.
3. При необходимости добавить к шагу запуска программы DD-операторы языка управления заданиями следует кодировать:
//GO.имя DD ...

4. Дополнительные возможности

4.1. Выделение разделов текстовой библиотеки (TXTGET)

```
// EXEC TXTGET ,LIB = имя библиотеки
[,L = 'список имен разделов' | SYSIN ]
[,M = S|E - способ выборки разделов]
//TXT.OUT DD UNIT=...,VOL=SER=...,SPACE=...,DISP=...
```

Примечания:

1. Процедура позволяет получить "чистые" /без управляющих операторов/ разделы текстовой библиотеки на каком-то внешнем носителе данных /карты, лента, диск/. Необходимое устройство задается в DD операторе с именем &&TXT. Разделы выводятся в алфавитном порядке.
2. M = S - выбираются только разделы из списка L.
= E - разделы из списка L не выбираются.
По умолчанию M = S и L = ALL, т.е. выводится вся библиотека полностью. Значение параметра L=SYSIN см. пункт 1.2, прим.7.
3. С помощью этой процедуры и любой из процедур пункта 3.1 можно протранслировать полностью всю библиотеку, например:
// EXEC TXTGET,LIB=HBOOK
// EXEC FTN,LIB=HBOOK
//MOD.SYSIN DD DSN=&&TXT,DISP=(OLD,DELETE)

4.2. Печать набора данных с нумерацией страниц и оглавлением (GDLIST)

```
// EXEC GDLIST [,M = C|T|D - способ печати]
```

Примечания:

1. M=T - печать всех модулей.
C - печать только комментариев перед телом модуля.
D - печать только оглавления. Принято по умолчанию.

2. Каждый модуль программы выносится на отдельный лист, страницы озаглавливаются и нумеруются; выдается также оглавление и список модулей в алфавитном порядке.
3. При выводе исходных модулей с ленты или диска необходимо при вызове процедуры добавить следующую карту:
//MOD.SYSIN DD ...

4.3. Работа с магнитными лентами (BUFFER)

Вызов из FORTRAN-программы:

```
.....
CALL BUFFER (ICOP, IFLAG, NBYTES, ARRAY, &1, &2, &3)
.....
END
//TAPEXX DD UNIT=..., VOL=SER=...
//TAPEYY DD UNIT=..., VOL=SER=...
и так далее но не более четырех лент.
```

Примечания:

1. Подпрограмма BUFFER предназначена для работы с магнитными лентами из программ, написанных на языках высокого уровня (FORTRAN, PLI). Подпрограмма позволяет осуществлять чтение и запись блоков переменной длины, обратное чтение блоков и операции управления для файлов на магнитной ленте.

Описание параметров:

ICOP - код производимых операций с магнитной лентой = XX*10+OP.
 XX - ссылочный номер набора данных. Указывается в DD-карте.
 OP=0 - чтение.
 1 - запись.
 2 - обратное чтение.
 3 - шаг назад на блок (BSB).
 4 - шаг вперед на файл (FSF).
 5 - запись ленточной марки (WTM).
 7 - перемотка ленты к точке загрузки (REW).
 8 - шаг вперед на блок (FSB).
 9 - уничтожить промежуток (ERG).
 IFLAG - задает число операций с магнитной лентой, содержит:
 а/ при чтении/записи - число переданных блоков.
 б/ при командах управления - число выполненных команд.

NBYTES - длина блока данных на ленте в байтах при вводе/выводе. (0<NBYTES<65355).
 ARRAY - массив для данных при вводе/выводе.
 &1, &2, &3 - метки возврата /могут быть опущены/:
 &1 - чтение ленточной марки.
 &2 - сбой в данных.
 &3 - сбой устройства.

Коды пользователя при аварийном завершении работы программы:

- 1/ 1000 - ошибочное имя DD - карты.
- 2/ 1001 - попытка подключения более четырех устройств.
- 3/ 1002 - ошибочный номер устройства.
- 4/ 1003 - ошибка при открытии набора данных на ленте.

4.4. Создание и настройка библиотеки процедур (CRPROC)

```
// EXEC CRPROC ,IND = индекс
,V = архивный номер диска
```

Примечания:

1. Действие процедуры обеспечивается утилитой IEBCOPY.
2. На управляющем томе пользователя создается библиотека с именем <индекс>,\$PROCTX, туда копируется содержимое библиотеки процедур SYS1,\$PROCTX и затем новая библиотека процедур настраивается на работу с новым индексом.

ОБЩИЕ ЗАМЕЧАНИЯ

1. При занесении раздела в библиотеку любого типа или его редактировании в элемент оглавления библиотеки, соответствующий этому разделу, записывается дата создания раздела в формате ГГММДДС.
2. В большинстве процедур по умолчанию используются библиотеки с именем \$COMN текстового и объектного типов. Это библиотеки общего назначения - макроопределения и часто используемые подпрограммы. Эти библиотеки должны находиться на управляющем или личном томе пользователя.
3. В процедурах ASM, ASMLD и ASMRUN реализована пакетная трансляция ассемблерных модулей.

ЗАКЛЮЧЕНИЕ

Опыт эксплуатации системы ведения личных библиотек на ЭВМ ЕС-1040 СНЭО ОИЯИ показал ее эффективность и простоту в эксплуатации.

Авторы выражают глубокую признательность Н.А.Воскресенскому за ряд предоставленных программ, Э.И.Мальцеву за интерес и внимание к настоящей работе, Л.П.Холоденко и Х.Эшер - за помощь в подготовке рукописи к печати.

ЛИТЕРАТУРА

1. Система математического обеспечения ЕС ЭВМ под общей редакцией А.М. Ларионова. "Статистика", М., 1974.
2. Лебедев В.Н., Соколов А.П. Введение в систему программирования ОС ЕС. "Статистика", М., 1978.
3. Митрофанов В.В., Одинцов В.В. Программы обслуживания ОС ЕС ЭВМ. "Статистика", М., 1977.
4. CERN Computer Centre IBM-370, Users Guide DD/US/4, 1978.
5. Организация личной библиотеки программ в ОС ЕС (OS/360) и работа с ней. Препринт ИАЭ-3163, М., 1979.
6. CERN 6000 Computer Program Library 2 (Z262), Geneva, 1979.

Рукопись поступила в издательский отдел
28 апреля 1980 года.