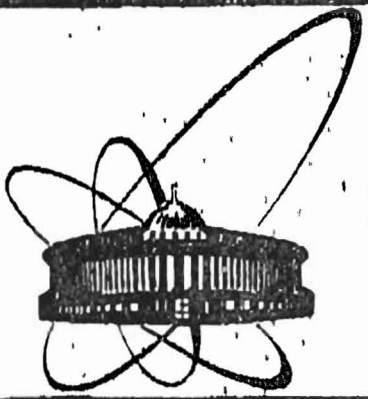
88-897



сообщения
объединенного
института
ядерных
исследований
дубна

Р11-88-897

А.М.Хасанов

EDITLIB/ARCHIVE – ПРОГРАММЫ ВЕДЕНИЯ АРХИВА ГЕНЕРАЦИИ БИБЛИОТЕКИ ЗАГРУЗОЧНЫХ МОДУЛЕЙ В ОС ЕС

1988

ВВЕДЕНИЕ

Каждый пользователь ОС ЕС постоянно имеет дело с библиотеками загрузочных модулей[1].

В них хранятся модули операционной системы, трансляторов, различных обрабатывающих программ, библиотек программ общего назначения. В библиотеках загрузочных модулей пользователи хранят свои собственные программы.

Обычно предназначенные для многократного использования программы транслируются и с помощью редактора связей IEWL записываются в библиотеки загрузочных модулей.

Редактор связей — многофункциональная программа, но она сложна в изучении и не всегда удобна в использовании. Пользователю обычно хватает нескольких возможностей редактора, но не всегда их легко реализовать.

Например, если нужно создать несколько загрузочных модулей, необходимо для каждого из них вызвать транслятор, редактор связей, указать в управляющих картах имя выходного загрузочного модуля. Когда таких модулей сотни или тысячи (как, например, в библиотеках общего назначения), то объем необходимой ручной работы становится очень большим.

Поэтому возникает задача автоматизировать некоторые процессы использования редактора связей. Для этих целей создана программа EDITLIB, которая динамически вызывает редактор связей IEWL посредством макрокоманды ATTACH. Она также создает и поддерживает архивы библиотеки загрузочных модулей.

Библиотеки загрузочных модулей иногда достигают очень больших размеров. В процессе их сопровождения многие модули исключаются, перезаписываются, добавляются новые. В одной библиотеке могут находиться модули, исходные программы которых транслированы разными компиляторами. Библиотека, таким образом, может превратиться в некий "черный ящик", и нельзя с уверенностью утверждать, что с ней все в порядке. При сопровождении таких библиотек возникают проблемы слежения за их состоянием, за тем, какие модули в них находятся, за их версиями, датами записи, за именами использованных трансляторов и т.п.

Программа EDITLIB является инструментом, позволяющим для любой библиотеки загрузочных модулей иметь некий автоматически поддерживаемый архив, в котором хранится вся информация о модулях библиотеки.

Главное, что от пользователя не требуется никаких усилий по поддержанию этого архива. Единственное, что нужно — пользоваться для записи загрузочных модулей в библиотеку программой EDITLIB. Программа сама создает архив, если его еще не было, модифицирует и сохраняет его.

Причем физически архив — обычный раздел той же библиотеки загрузочных модулей, информацию о которой он хранит. Таким образом, с пользователя снимается забота о создании файлов для хранения архива. При копировании библиотеки архив также автоматически копируется как раздел библиотеки.

В любой момент времени архив содержит полную информацию о состоянии библиотеки, которую может распечатать программа ARCHIVE.

ПОНЯТИЕ МОДУЛЯ, ПРОСТОГО МОДУЛЯ И МУЛЬТИМОДУЛЯ

Обычно программы пишутся с использованием метода модульного программирования. Большая единая задача делится на подзадачи и, соответственно, программа разбивается на логические части, каждая из которых может быть закодирована на языке программирования, наиболее подходящем для нее, и затем раздельно ассемблирована или компилирована соответствующим транслятором[2]. Такие логические части программы называются модулями.

Таким образом, исходный модуль — относительно независимая программная единица, как, например, процедура в Алголе, подпрограмма или подпрограмма-функция в Фортране, раздел в Коболе. Будем считать именем модуля имя такой программной единицы. Для ассемблера ЕС ЭВМ под модулем понимаем секционированную программу, состоящую из одной или нескольких секций и заканчивающуюся псевдокомандой END[3]. Имя первой или единственной псевдокоманды START или CSECT определяет имя модуля.

Исходные модули компилируются соответствующим транслятором, на выходе получается двоичный код, который называется объектным кодом или объектным модулем. Объектный модуль после обработки редактором связей превращается в загрузочный модуль, который записывается в библиотеку загрузочных модулей.

Каждый модуль может содержать символические ссылки на объекты в других модулях. Такие ссылки называются внешними.

Загрузочный модуль, полученный из одного объектного модуля, будем называть простым. Загрузочный модуль, созданный из нескольких объектных и загрузочных модулей, назовем мультимодулем.

Простые загрузочные модули создаются без разрешения внешних ссылок. В таком виде хранятся подпрограммы, которые часто используются в других программах для решения какой-то частной подзадачи. Это экономит место и предотвращает двойное определение имен. Для использования таких модулей их нужно еще раз обработать редактором связей. Мультимодули с неразрешенными ссылками также должны обрабатываться редакторами перед использованием. Отлаженные программы, полностью решающие некоторую задачу, хранятся в виде мультимодулей с разрешенными ссылками. Они занимают больше места, но при этом экономится время, так как такие модули можно сразу выполнять без редактирования.

ПРОГРАММА EDITLIB

Одной из функций программы EDITLIB является функция обработки объектного файла и записи загрузочных модулей в библиотеку.

Объектный файл — последовательный файл, состоящий из одного или нескольких объектных модулей, полученных одним или различными трансляторами.

Будем называть сеансом записи каждое обращение к программе EDITLIB. Для любого сеанса должен быть задан режим записи: запись простых модулей или запись мультимодулей.

В режиме записи простых модулей каждый объектный модуль обрабатывается отдельно и в библиотеку записывается соответствующий ему загрузочный модуль под соответствующим именем. В этом режиме программа EDITLIB может записать в библиотеку любое количество простых модулей. Во втором режиме все объектные модули обрабатываются вместе и в библиотеку записывается один мультимодуль под именем, заданным пользователем.

Независимо от режима записи из каждого объектного модуля извлекается информация, которая затем хранится в архиве библиотеки.

Объектные модули, полученные любым транслятором ОС ЕС, имеют одинаковую структуру и состоят из записей четырех типов: ESD, TXT, RLD и END[3].

ESD (External Symbol Dictionary) — словарь, содержащий описание всех внешних символов модуля. Перечислим возможные типы внешних символов:

SD (Section Definition) — имя программной секции;
LD (Link Definition) — дополнительное имя;
ER (External Reference) — внешняя ссылка;
PC (Partition Code) — частный код;
CM (CoMmon area) — общая область;
PR (Pseudo Register) — псевдорегистр;
WX (Weak eXternal) — слабая внешняя ссылка.

Ассемблер может создавать все типы внешних символов, кроме поименованных общих блоков; FORTRAN — все, кроме PC, PR, WX; COBOL — все, кроме CM и PR; PL/1 — все, кроме непоименованных общих областей.

Для символов типа SD, PC, CM и PR в словаре содержится также соответствующая им длина.

Из этого словаря выбирается для последующего сохранения в архиве следующая информация о внешних символах модуля: имя символа, его тип и, если есть, длина. Имя первого символа типа SD принимается за имя модуля. В режиме записи простых модулей под этим именем загрузочный модуль записывается в библиотеку, то есть это имя используется в управляющем предложении NAME редактора связей IEWL, а имена символов типа LD используются в управляющих предложениях ALIAS и становятся дополнительными именами загрузочного модуля в библиотеке.

В режиме записи мультимодуля в предложении NAME используется имя, которое пользователь задал для мультимодуля.

В обоих режимах добавляется управляющее предложение SETSSI, которое заставляет редактор связей записать в элемент оглавления для модуля 4-байтовое поле с датой его записи. Заметим, что существует программа (и соответствующая процедура) ADICT, которая выдает содержимое оглавления библиотеки с датой записи каждого модуля.

Объектный модуль заканчивается записью типа END, которая служит программе EDITLIB для выделения отдельных модулей. Из этой записи также выбирается имя, версия и модификация транслятора, который применялся для компилирования данного модуля. Эта информация также сохраняется в архиве. Приведем обозначения некоторых известных трансляторов в объектных модулях:

```
FORTRAN H Extended : 5734.F03    0202
FORTRAN VS(V 1.3) : 5748.F03    0100
FORTRAN VS(V 2.0) : 5668-806    0200
FORTRAN VS(V 2.1) : 5668962. 010201
FORTRAN Fujitsu   : JZK          1020
Ассемблер H       : 5734AS100  0501
Ассемблер F       : 52ASM316860001
Pascal-8000       : PASCAL 8000/2
```

Отладочный транслятор FORTRAN G и оптимизирующий FORTRAN H не помещают своего имени в запись END.

Информация о трансляторе очень полезна, так как позволяет четко представлять, какие трансляторы использовались при формировании библиотеки загрузочных модулей.

Некоторые сложности возникают с программной единицей FORTRAN'a BLOCK DATA, которая оформляется транслятором как отдельный объектный модуль. Именам общих блоков в нем присваивается тип SD, а не CM. Если обрабатывать такой модуль как обычный и записывать его в библиотеку под одним из этих имен, то в дальнейшем его нельзя будет подсоединить к программе, так как обращений к этому имени другие модули не содержат.

Поэтому программой EDITLIB такой модуль отдельно не записывается, а объединяется со следующим за ним модулем. В связи с этим важно при трансляции модуль BLOCK DATA помещать где-нибудь в середине программы и не давать ему имени, иначе он не будет распознаваться программой EDITLIB.

INDEX-ФАЙЛЫ

Программе, решающей некоторую задачу и состоящей из нескольких модулей, обычно присваивается некоторое произвольно выбранное имя. Это может быть, например, имя головного модуля программы. В библиотеке "Дубна"[4] это индекс программы. В дальнейшем для единообразия будем использовать термин "индекс программы".

Если в библиотеке должны храниться модули, относящиеся к разным программам, то при использовании программ EDITLIB и ARCHIVE есть возможность задавать разбиение модулей по индексам. Заметим, что это разбиение является формальным и не касается способа записи и хранения модулей в библиотеке. Оно относится только к организации архива библиотеки, при которой вся информация о модулях одного индекса обрабатывается, хранится и выдается совместно, что значительно упрощает и облегчает сопровождение больших библиотек загрузочных модулей.

Для задания разбиения модулей по индексам введено понятие INDEX-файла. Его использование необязательно, тогда считается, что все модули библиотеки относятся к специально выделенному BLANK-индексу.

Структура INDEX-файла:

```
%INDEX ⟨index1⟩
%INCLUDE ⟨module11⟩ ⟨module12⟩ . . .
%⟨module13⟩
    ⋮
%INDEX ⟨index2⟩
%1 ⟨module21⟩ ...
    ⋮
```

Предложение %INDEX задает имя индекса, а следующие за ним предложения %INCLUDE задают имена модулей, принадлежащих к этому индексу. Вместо %INCLUDE можно использовать %I. В одной строке %I до 72-й позиции включительно можно записать несколько имен, разделенных любым из символов: пробел, точка, запятая.

Если INDEX1 не задан, то все модули до следующего индекса относятся к специально выделенному BLANK-индексу. Модули, не включенные ни в один из индексов, также относятся к BLANK-индексу. Частный случай этого — когда INDEX-файл вообще не используется.

Символ % в предложениях INDEX и INCLUDE означает, что индексу или модулю приписывается статус "поддерживается". Символ # озна-

чает статус "не поддерживается", которым отмечаются устаревшие или не используемые индексы или модули. Если некоторый индекс имеет статус "не поддерживается", то и все его модули автоматически получают такой же статус, даже если они заданы предложением %I. Присвоение статуса модулям и индексам также играет чисто формальную роль.

СТРУКТУРА АРХИВА

Напомним, что архив, создаваемый программой EDITLIB, является разделом библиотеки загрузочных модулей, информацию о которой он хранит. Имя этого раздела #ARCHIVE. Он состоит из записей формата U, длины, равной длине блока библиотеки. Лишь последняя запись раздела может быть короче.

Чтение и запись раздела #ARCHIVE осуществляется программами комплекса, описанного в[5]. Архив имеет определенную внутреннюю логическую структуру (см. приложение). Он состоит из пяти основных частей, которые, в свою очередь, делятся на более мелкие элементы данных. Каждый элемент данных начинается с определенного символа (IDENT), который идентифицирует его тип:

— HEADER удостоверяет, что данный раздел является архивом и, кроме того, в нем находятся такие данные, как количество элементов COMP-data в COMPLIST и RUN-data в RUNLIST, а также версия программы EDITLIB, записавшей этот архив;

— COMPLIST содержит фирменные названия, номера версий и модификации всех трансляторов, которые использовались для трансляции программ, записанных в библиотеку;

— RUNLIST содержит описания всех состоявшихся сеансов записи в библиотеку. Для каждого сеанса хранятся дата и время сеанса, количество модулей, записанных в библиотеку за сеанс, а также данные пользователя, в которых он может давать характеристику сеанса;

— MODLIST состоит из элементов данных трех типов, которые могут быть вложены друг в друга: INDEX-data, MODULE-data, SYMBOL-data;

— TAIL состоит из одного символа IDENT и специфицирует конец архива.

INDEX-data определяет индекс, к которому относятся все следующие элементы MODULE-data. Здесь также записаны статус индекса и количества модулей в индексе.

MODULE-data содержит общую информацию о модуле, записанном в библиотеку: его имя, статус, длину в байтах, номер RUN-data в RUNLIST и COMP-data в COMPLIST. Номер RUN-data определяет сеанс, в котором этот модуль был записан в библиотеку. Зная его, мы можем узнать дату и время записи. По номеру COMP-data определяется имя транслятора, которым транслировался модуль.

Внешние символы модуля всех типов, кроме SD, описаны в SYMBOL-data. Здесь задается имя символа (или пробелы) и тип символа. Для символов типа PC, CM и PR задается также соответствующая ему длина. Такие символы имеют IDENT "+".

В архиве содержится информация только о тех модулях, которые записаны (без ошибок) в библиотеку. Соответственно, элемент INDEX-data помещается в архив, если есть хотя бы один относящийся к нему элемент MODULE-data.

ОБНОВЛЕНИЕ И ЗАПИСЬ АРХИВА

Итак, во время обработки объектного файла происходит не только редактирование и запись модулей в библиотеку, но и сбор информации о данном сеанса. Эта информация накапливается в оперативной памяти, причем данные о каждом модуле — под соответствующим ему индексом. По окончании обработки объектного файла описание данного сеанса выводится на печать в том же формате, как это делает программа ARCHIVE (см. ниже).

На следующем этапе информация о текущем сеансе сливается с предыдущим вариантом архива (если он существовал) из раздела #ARCHIVE библиотеки загрузочных модулей. Слияние происходит также в памяти, причем информация, полученная во время данного сеанса, имеет приоритет перед старой информацией из архива. Данные о вновь записанных модулях заменяют предыдущие данные об одноименных модулях. Список трансляторов дополняется именами трансляторов, впервые использованных в этом сеансе.

В зависимости от INDEX-файла модули могут перемещаться от одного индекса к другому, индексы и модули могут менять свой статус. INDEX-файл не обязательно должен содержать имена всех модулей, которые записаны в библиотеку. Если в старом архиве есть индексы и модули, которые не описаны в INDEX-файле, то они не подвергаются никаким изменениям.

INDEX-файл, следовательно, можно применять для перераспределения модулей по индексам, для изменения их статуса. Можно, например, несколько сеансов записи провести, не используя INDEX-файлов, тогда все модули записываются под BLANK-индексом. Когда же формирование библиотеки закончено, можно распределить модули по нужным индексам, задав соответствующий INDEX-файл.

После завершения формирования нового архива в памяти он записывается в библиотеку в раздел #ARCHIVE, заменяя существовавший раздел с таким именем.

Таким образом, при каждом сеансе записи в библиотеку автоматически производится обновление архива библиотеки в соответствии с последними изменениями в ней.

ПРОГРАММА ARCHIVE

Как только архив создан и записан в раздел #ARCHIVE библиотеки загрузочных модулей, можно выдать на печать его содержимое, то есть фактически всю информацию о текущем состоянии библиотеки, с помощью программы ARCHIVE. Здесь также возможно использование INDEX-файла, который выполняет те же функции, что и в программе EDITLIB. Например, если до этого ни разу не применялся INDEX-файл, то есть все модули приписаны к BLANK-индексу, можно сделать выдачу архива упорядоченной по индексам. Если INDEX-файл не задан, то архив выдается с таким разбиением по индексам, как в нем зафиксировано.

Вначале из раздела RUNLIST архива выдается краткая информация о всех сеансах записи в библиотеку: номер сеанса, дата и время записи, а также данные пользователя.

Затем выдается информация из раздела COMPLIST. Это номер транслятора для дальнейшей ссылки на него, имя, версия и модификация транслятора.

И, наконец, из раздела MODLIST для каждого модуля выдается его имя, длина в байтах, номер компилятора из раздела COMPLIST, номер сеанса из RUNLIST и имена внешних символов модуля, кроме типа SD. Для символов типа CM, PR и PC указывается также соответствующая им длина. Статус модуля отмечается символом перед его именем: пробел — статус "поддерживается", минус — "не поддерживается".

Для мультимодуля сначала выдается его имя, длина и номер сеанса, затем данные о всех модулях, из которых он состоит, помеченные символом "звездочка" (*).

Возможны несколько режимов выдачи. В режиме "A" модули выдаются под своими индексами, причем индексы упорядочены по алфавиту. Индексы, имеющие статус "не поддерживаются", отмечаются символом "—".

В режиме "B" модули выдаются в алфавитном порядке с указанием индекса, к которому принадлежит модуль. Режимы "AX" и "BX" уточняют режимы "A" и "B" и действуют только при заданном INDEX-файле. Если в режимах "A" и "B" индексы и имена модулей из INDEX-файла выдаются только в том случае, когда либо они есть в архиве, либо имеют статус "поддерживается", то в режимах "AX" и "BX" они выдаются всегда.

ПРОЦЕДУРЫ EDITLIB И ARCHIVE

Для использования программ EDITLIB и ARCHIVE заведены одноименные процедуры. Процедура EDITLIB имеет следующие параметры управления. Указаны значения параметров, заданных в процедуре:

RG = 400K	Объем памяти. Если модулей много (больше 1000), то память надо увеличить.
TIM = 5	Для записи 300-400 модулей обычно хватает трех минут процессорного времени на EC-1061.
A = Y	Архив создавать. При отказе от ведения архива задать A = N.
C = N	Редактировать без реализации внешних ссылок (параметр NCAL в редакторе связей). C = Y — редактирование с реализацией внешних ссылок, тогда надо задать и соответствующие библиотеки (см. ниже).
N =	Задает режим записи простых модулей. Для записи мультиполя следует указать его имя N = name.
T =	Данные пользователя длиной от 0 до 60 символов в кавычках. Нельзя использовать символы "/" и ",".
LIBDSN =	Имя библиотеки загрузочных модулей.
LIBVOL =	Имя тома, если библиотека не каталогизирована или создается в данном сеансе.
LIBDSP = ` OLD´	Диспозиция для существующей библиотеки. Если библиотека создается в этом сеансе, то LIBDSP = ´(,KEEP)´. Новая библиотека будет иметь DCB = (RECFM = U, BLKSIZE= = 7294) и SPACE = (CYL,(15,5,50)).
INDDSN = NULLFILE	По умолчанию INDEX-файл не используется.
INDVOL =	Имя тома с INDEX-файлом.
OBJDSN = ´&LOADSET´	По умолчанию используется файл с объектными модулями, который создает транслятор на предыдущем шаге.
OBJVOL =	Имя тома с объектным файлом.
LLB1 = ´SYS1.DUMMYL´	Если C = Y, то здесь задаются имена библио-
LLB2 = ´SYS1.DUMMYL´	тек для реализации внешних ссылок.
LPRN = A	Класс вывода для редактора связей.
APRN = A	Класс вывода для распечатки архива. Если APRN = S, то выдача разделяется на страницы.

Процедура ARCHIVE имеет следующие параметры, аналогичные параметрам процедуры EDITLIB: RG, TIM, LIBDSN, LIBVOL, INDDSN, INDVOL, LPRN, APRN, а также параметр TYPE, который задает режим выдачи. По умолчанию используется режим "A".

ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ПРОЦЕДУР

1. //EXEC EDITLIB, LIBDSN=NEWLIB, LIBVOL=' USRVOL,
   //LIBDSP='(,CATLG)', OBJDSN=OBJFILE

Объектные модули из файла OBJFILE редактируются и записываются в новую библиотеку загрузочных модулей в отдельные разделы по именам модулей без разрешения ссылок. Создается архив, который записывается в раздел #ARCHIVE библиотеки.

2. // EXEC FOR77C
   // SYSIN DD DSN=FOR77.FILE, DISP=SHR
   // EXEC EDITLIB, LIBDSN=NEWLIB, T='SYSTEM ONE V 3.1'

Файл исходных модулей транслируется, загрузочные модули записываются в NEWLIB, архив обновляется.

3. // EXEC FORTXC
   // SYSIN DD DSN=MINUIT, DISP=SHR
   // EXEC EDITLIB, LIBDSN=NEWLIB, N=MINTLD,
   // T='MINUIT DOUBLE PRECISION V5.0'

Программа транслируется и записывается в библиотеку как мультимодуль с именем MINTLD без разрешения внешних ссылок. Все модули, записанные в библиотеку в приведенных выше примерах, могут быть подключены в дальнейшем с помощью оператора CALL и подсоединения библиотеки NEWLIB на шаге редактирования связей.

4. // EXEC YPATCHY, PAMDSN=PAM1
   +USE.
   +PAM.
   +ASM,22
   +EXE.
   +QUIT.
   // EXEC FOR77C
   // SYSIN DD DSN=&ASM, DISP=OLD
   // EXEC ASMA
   // EXEC EDITLIB, LIBDSN=NEWLIB, N=SUPER,
   // C=Y, INDDSN='LIBR(IND)',
   // LLB1=LIB1, LLB2=LIB2,
   // T='SUPER1.0 РАСЧЕТ МАГНИТНЫХ ПОЛЕЙ'

Полностью отлаженная программа выбирается из PAM-файла PAM1. Транслируются модули на FORTRAN-77, записываются во временный файл &LOADSET. Затем используется программа ASMA, которая выполняет трансляцию всех модулей на ассемблере. Полученные объектные модули добавляются в тот же файл &LOADSET, который редактируется с использованием библиотек LIB1 и LIB2 и записывается в библиотеку NEWLIB единым модулем с именем SUPER. Этот модуль в дальнейшем можно вызывать в виде:

   // EXEC PGM=SUPER
   // STEPLIB DD DSN=NEWLIB, DISP=SHR

5. // EXEC ARCHIVE, LIBDSN=NEWLIB, APRN=S

Архив библиотеки NEWLIB выводится в режиме "A" на форматную печать в класс S.

6. // EXEC ARCHIVE, LIBDSN=NEWLIB, TYPE=BX,
   // INDDSN='LIBR(IND)'

Распечатывается в режиме "BX" архив библиотеки NEWLIB с использованием INDEX-файла, который берется из раздела IND текстовой библиотеки LIBR.

ЗАМЕЧАНИЯ

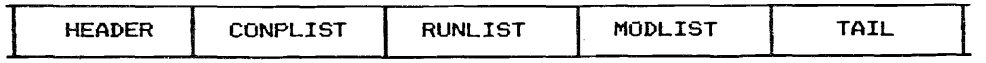
При использовании программы EDITLIB необходимо придерживаться определенной дисциплины.

Для того чтобы архив всегда точно отражал состояние библиотеки, запись загрузочных модулей в библиотеку должна производиться только программой EDITLIB. Модули, записанные в библиотеку другими программами, должны быть перезаписаны с использованием EDITLIB.

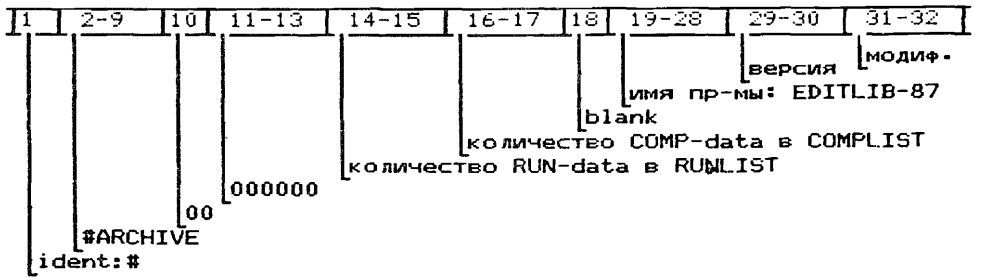
Если сеанс записи окончился преждевременно (по зависанию системы, нехватке места, по "cancel" и т.д.), то его требуется повторить заново, т.к. в этом случае архив уже не отражает достаточно верно состояние библиотеки. Если библиотека уже заполнена, необходимо предварительно расширить ее или сделать ей компрессию.

Приложение

#ARCHIVE:

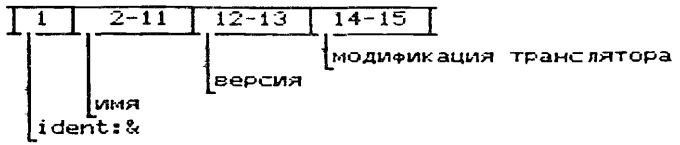


HEADER:

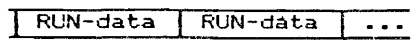


COMPLIST:

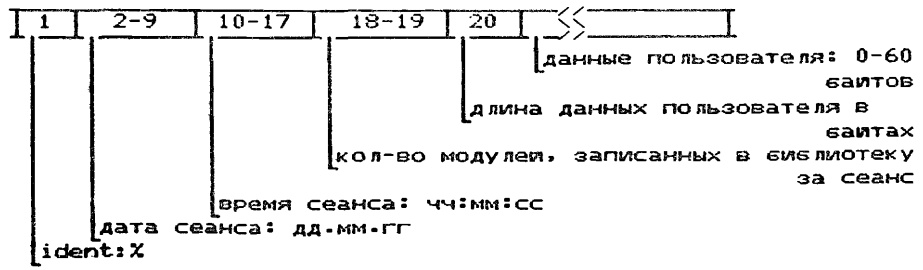
COMP-data:



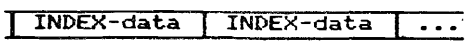
RUNLIST:



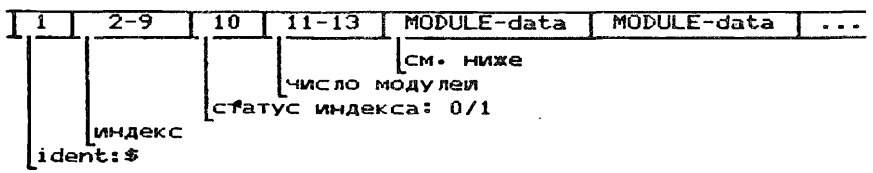
RUN-data:



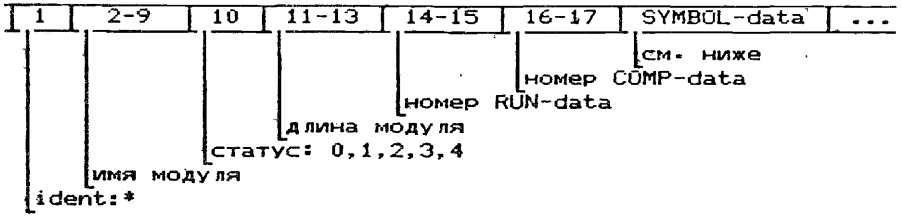
MODLIST: (min: 1 INDEX-data)



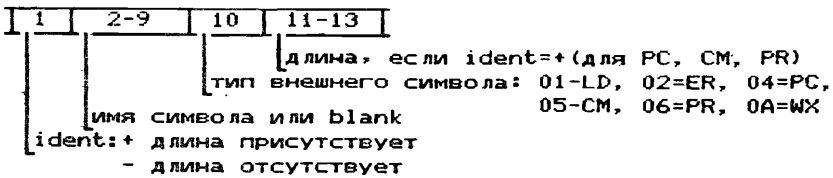
INDEX-data:



MODULE-data: (min: 17 байтов)



SYMBOL-data: (min: 10 байтов, max: 13 байтов)



TAIL:

ЛИТЕРАТУРА

1. Тимонин В.И. — Операционная система ОС ЕС. Основы функционирования. М.: Финансы и статистика, 1983.
2. Ван Тассел Д. — Стиль, разработка, эффективность, отладка и испытание программ. М.: Мир, 1981.
3. Зайцева Ж.Н.— Программирование в ОС ЕС на базе языка ассемблера. М.: Финансы и статистика, 1981.
4. Лукстиня Л. и др. — ОИЯИ, Р11-85-170, Дубна, 1985.
5. Хасанов А.М. — Сообщение ОИЯИ Р11-87-584, Дубна, 1987.

Рукопись поступила в издательский отдел
27 декабря 1988 года.