

1031/2-48

СООБЩЕНИЯ  
ОБЪЕДИНЕННОГО  
ИНСТИТУТА  
ЯДЕРНЫХ  
ИССЛЕДОВАНИЙ

ДУБНА

27/11-78



11 - 11134

В.В.Галактионов, В.В.Кореньков

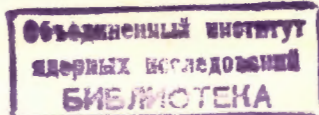
ОБРАБОТКА ТЕКСТОВ  
ПРОГРАММ НА ФОРТРАНЕ  
ИЗ ПОСЛЕДОВАТЕЛЬНОГО НАБОРА ДАННЫХ  
НА ЕС ЭВМ В СИСТЕМЕ ОС

1977

11 - 11134

В.В.Галактионов, В.В.Кореньков

ОБРАБОТКА ТЕКСТОВ  
ПРОГРАММ НА ФОРТРАНЕ  
ИЗ ПОСЛЕДОВАТЕЛЬНОГО НАБОРА ДАННЫХ  
НА ЕС ЭВМ В СИСТЕМЕ ОС



Галактионов В.В., Кореньков В.В.

11 - 11134

Обработка текстов программ на ФОРТРАНе из последовательного набора данных на ЕС ЭВМ в системе ОС

В работе описываются функции, возможности и методика использования программы UPFORTXT, предназначенной для обработки на ЕС ЭВМ в системе ОС текстов программ на языке ФОРТРАН, образующих последовательный набор данных.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1977

Galaktionov V.V., Korenkov V.V.

11 - 11134

Updating of FORTRAN Program Texts Forming a Sequential Data Set with the Computer ES in OS System

Functions, possibilities and methods are described of using UPFORTXT program intended to updating FORTRAN program texts forming a sequential data set with the computer ES in OS system.

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

Communication of the Joint Institute for Nuclear Research. Dubna 1977

Программа UPFORTXT предназначена для обработки текстов программ на языке ФОРТРАН, образующих последовательный набор данных, в частности на магнитной ленте, что создает определенные трудности при работе с текстовыми наборами данных большого объема, состоящими из множества подпрограмм. В частности, такая задача возникла в связи с переносом библиотеки программ с больших ЭВМ БЭСМ-6 и СДС-6500 на ЭВМ ЕС-1040 и формированием библиотеки программ в системе ОС на ЭВМ ЕС-1040.

На ЕС ЭВМ в рамках операционной системы ОС нет в готовом виде аппарата для создания библиотечных наборов данных с магнитных лент, а работа с последовательными наборами данных большого объема, записанными на магнитной ленте, затруднена.

Программа UPFORTXT, в частности, значительно облегчает работу по формированию библиотечного набора данных с магнитных лент и осуществляет:

- просмотр последовательного набора данных, распечатку первой и последней записи каждого программного модуля;
- распечатку первой и последней записи, имеющих одинаковый индекс (содержимое 73-76-й позиций каждой записи);
- выборку из исходного набора данных программных модулей с заданными именами или индексами и формирование из них нового набора данных;
- выборку из текстового набора данных любого фрагмента и формирование нового набора данных;
- формирование промежуточного набора данных из исходного, который легко можно использовать для образования библиотечного набора данных с помощью служебной программы IEBURDTE.

Каждый программный модуль на языке ФОРТРАН начинается с операторов <тип> FUNCTION или SUBROUTINE и кончается оператором END. Каждая запись на магнитной ленте представляет собой запись карточного формата длиной в 80 байтов, где в 73-80 позициях указан номер записи, причем 73-76 позиции содержат индекс записи.

Обычно программные модули, связанные между собой или выполняющие одинаковые функции, записываются под одним индексом. Записи могут быть заблокированы с любым коэффициентом блокировки, допустимым в ОС. Программа UPFORTXT написана на АСМЕЛЕРЕ в рамках операционной системы ОС на ЕС-1040.

Программа UPFORTXT использовалась при переносе библиотеки программ с БЭСМ-6 на ЕС-1040.

#### ФУНКЦИИ ПРОГРАММЫ UPFORTXT

Программа UPFORTXT выполняет семь различных функций, которые задаются в поле PARM управляющего оператора EXEC.

1) LIST NAMES - распечатка первой и последней записи каждого программного модуля, находящегося на магнитной ленте.

Задание функции:

```
//_ EXEC _ PGM=UPFORTXT, PARM='LIST _ NAMES'
```

Например, для набора данных, состоящего из четырех программных модулей, распечатка будет иметь следующий вид:

```
SUBROUTINE PARINT(X,X1,N,F,N,R)      E1100001
END                                     E1100037
COMPLEX FUNCTION FCN(Z,I)            C203T011
END                                     C203T015
SUBROUTINE NZEROS(NZ,Z,R,FCN)        C2030002
END                                     C2030085
FUNCTION NEAR1(X)                     D3110011
END                                     D3110152
```

\*\*\* NO ANY ROUTINE ON TAPE

2) LIST INDEX - распечатка первой и последней записи текстов программ с одним и тем же индексом (позиции 73-76 в каждой записи).

Задание функции:

```
//_ EXEC _ PGM = UPFORTXT, PARM = 'LIST _ INDEX'
```

Для описанного выше набора данных распечатка будет иметь следующий вид:

```
SUBROUTINE PARINT(X,X1,N,F,N,R)      E1100001
END                                     E1100037
COMPLEX FUNCTION FCN(Z,I)            C203T011
END                                     C2030085
FUNCTION NEAR1(X)                     D3110011
END                                     D3110152
```

В данном случае два программных модуля, FCN и NZEROS, имеют один и тот же индекс: C 203.

3) TAKE NAMES - выбрать из исходного текстового набора данных программные модули с заданными именами и образовать из них новый набор данных.

Задание функции:

```
//_ EXEC _ PGM = UPFORTXT, PARM = 'TAKE _ NAMES:<ИМЯ 1>, <ИМЯ 2>,
..., <ИМЯ n>'
```

Количество программных модулей, которые можно выбрать из текстового набора данных за один шаг задания, ограничивается длиной поля PARM. Максимальная длина поля PARM - 100 байтов, поэтому сумма всех символов (вместе с запятыми) не должна превышать 90.

Например, требуется выбрать из набора данных, описанного выше, два программных модуля с именами FCN и NEAR1 и образовать из них новый набор данных. Для этого случая функция задается следующим образом:

```
//_ EXEC _ PGM = UPFORTXT, PARM='TAKE _ NAMES:FCN,NEAR1'
```

После удачного завершения работы распечатка будет иметь следующий вид:

```
FOUND ROUTINE FCN
FOUND ROUTINE NEAR1
*** ALL ROUTINES ARE FOUND
```

Эта возможность, в частности, может использоваться для выбора и трансляции отдельных программных модулей из последовательного набора данных.

4) TAKE INDEX - выбрать из исходного текстового набора данных программные модули с заданными индексами и образовать из них новый набор данных.

Задание функции:

```
//_ EXEC _ PGM =UPFORTEXT,PARM='TAKE _INDEX:< индекс 1 > ,< индекс 2 > ,  
... , < индекс n >'
```

Каждый индекс имеет фиксированную длину - 4 байта, причем ни один из четырех символов не может быть пробелом (см. замечание I)\*.

Максимально за один шаг задания можно выбрать программные модули, соответствующие восемнадцати индексам, и образовать из них новый набор данных. Например, требуется выбрать из описанного выше набора данных модули с индексами C 203 и D311 и образовать из них новый набор данных.

Задание функции для данного случая:

```
//_ EXEC _ PGM=UPFORTEXT,PARM='TAKE _INDEX:C203,D311'
```

После выборки и образования нового набора данных распечатка будет иметь следующий вид:

```
*** WRITE C203  
*** WRITE D311
```

5) TAKE SEGMT - выбрать из исходного набора данных часть текста и образовать из него новый набор данных.

Задание функции:

```
// EXEC PGM=UPFORTEXT,PARM='TAKE _SEGMT:<номер 1 > , <номер 2 >'
```

Текст выбирается, начиная с записи, имеющей <номер 1> в позициях 73-80, и кончая записью, имеющей <номер 2> в позициях 73-80.

Например, требуется выбрать из исходного набора данных часть текста, начиная с записи C 2030011 и кончая записью D3110152.

В этом случае функция задается таким образом:

```
//_ EXEC _ PGM=UPFORTEXT,PARM='TAKE _SEGMT:C2030011,D3110152'
```

\* Замечание I. Если хотя бы один символ индекса есть пробел, то запись считается принадлежащей модулю с предшествующим индексом, т.е. неправильные индексы при выборке игнорируются. Это замечание касается функции TAKE INDEX и LIBR INDEX.

После выборки части текста и образования нового набора данных программа выдает сообщение:

```
*** WRITE SEGM C2030011: D3110152
```

6) LIBR NAMES - получение из исходного набора данных промежуточного, который можно использовать для образования библиотечного набора данных по именам программных модулей. При этом в промежуточном наборе данных перед каждым программным модулем вставляется запись:

```
./_ ADD _ NAME = < имя модуля >
```

В результате этого можно легко образовать библиотечный набор данных с помощью системной служебной программы IEVURDTE. Указанный промежуточный набор данных будет входным набором для программы

IEVURDTE .

Задание функции:

```
//_ EXEC _ PGM =UPFORTEXT,PARM='LIBR _NAMES'
```

Для набора данных, используемого в предыдущих примерах, распечатка будет иметь следующий вид:

```
./_ ADD _ NAME=PARINT _ _ _ _ _ - вставляемая запись  
./_ ADD _ NAME=FCN _ _ _ _ _ - вставляемая запись  
./_ ADD _ NAME=NZEROS _ _ _ _ _ - вставляемая запись  
./_ ADD _ NAME=NEAR1 _ _ _ _ _ - вставляемая запись  
*** NO ANY ROUTINE ON TAPE
```

7) LIBR INDEX - получение из текстового исходного набора данных промежуточного, который можно использовать для образования библиотечного набора данных по именам индексов.

В промежуточном наборе данных перед программными модулями с одинаковыми индексами вставляется запись:

```
./_ ADD _ NAME= < имя индекса >
```

В результате этого можно легко образовать библиотечный набор данных с помощью системной служебной программы IEVURDTE, причем каждый раздел этого библиотечного набора данных состоит из программных модулей с одним индексом.

Указанный промежуточный набор данных будет входным набором для программы IEVURDTE .

Задание функции:

```
// EXEC PGM=UPFORTXT, PARM='LIBR INDEX'
```

Эта функция использует только правильные индексы, т.е. индексы, не содержащие пробелов (см. замечание 1).

Распечатка для рассматриваемого выше набора данных будет иметь следующий вид:

```
./ ADD NAME = E110 - вставляемая запись  
./ ADD NAME = C203 - вставляемая запись  
./ ADD NAME = D311 - вставляемая запись
```

#### Примеры использования программы UPFORTXT

1) Пусть имеется магнитная лента с текстами подпрограмм на языке ФОРТРАН. Нужно узнать, какие подпрограммы записаны, не распечатывая содержимого всей магнитной ленты.

Управляющие карты такого задания:

```
// LISNAM JOB CLASS=F,MSGLEVEL=1  
// STEP1 EXEC PGM=UPFORTXT, PARM='LIST NAMES'  
// SYSPRINT DD SYSOUT=A, DCB=(LRECL=80, BLKSIZE=80)  
// RDTAPE DD DSN=FORTEXT, UNIT=5010, DISP=(OLD, KEEP),  
// VOLUME=SER=TAPE01, LABEL=(2, SL)  
// DISC DD DSN=FORINP, DISP=(, PASS), UNIT=5050,  
// SPACE=(TRK, (10, 2)),  
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=800), VOL=SER=111111  
//
```

2) С магнитной ленты, содержащей тексты фортрановских подпрограмм, выбрать программный модуль с именем NZEROS, пропустить через транслятор и редактор связей, а затем записать загрузочный модуль в библиотечный набор данных SYS1.FORTLIB.

```
// LIBFOR JOB CLASS=F,MSGLEVEL=1  
// STEP1 EXEC PGM=UPFORTXT, PARM='TAKE NAMES:NZEROS'  
// SYSPRINT DD SYSOUT=A, DCB=(LRECL=80, BLKSIZE=80)  
// RDTAPE DD DSN=FORTEXT, UNIT=5010, DISP=(OLD, KEEP),  
// VOLUME=SER=TAPE01, LABEL=(2, SL)  
// DISC DD DSN=FORINP, DISP=(, PASS), UNIT=5050,  
// SPACE=(TRK, (10, 2)),  
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=800), VOL=SER=111111  
// STEP2 EXEC PROC=FORIGCL  
// FORT1.SYSIN DD DSN=FORINP, DISP=(OLD, DELETE)  
// LKEY.SYSLMOD DD DSN=SYS1.FORTLIB(NZEROS), DISP=OLD  
//
```

3) Образовать библиотечный набор данных FORLIB на диске SYSLIB по индексам из исходного последовательного набора данных на НМЛ с текстами программ на ФОРТРАНЕ.

```
// LIBRAR JOB CLASS=C,MSGLEVEL=1  
// STEP1 EXEC PGM=UPFORTXT, PARM='LIBR INDEX'  
// SYSPRINT DD SYSOUT=A, DCB=(LRECL=80, BLKSIZE=80)  
// RDTAPE DD DSN=FORTEXT, UNIT=5010, DISP=(OLD, KEEP),  
// VOLUME=SER=TAPE01, LABEL=(2, SL)  
// DISC DD DSN=FORINP, DISP=(, PASS), UNIT=5050,  
// SPACE=(TRK, (10, 2)),  
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=800), VOL=SER=111111  
// STEP2 EXEC PGM=IEBUPDTE, PARM=NEW  
// SYSPRINT DD SYSOUT=A  
// SYSUT2 DD DSN=FORLIB, UNIT=5050, DISP=(NEW, CATLG),  
// VOL=SER=SYSLIB,  
// SPACE=(CYL, (10, 2, 20)),  
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=800)  
// SYSIN DD DSN=FORINP, DISP=(OLD, DELETE)  
//
```

#### ЛИТЕРАТУРА

1. Г. Катцан. Операционные системы. Изд-во "Мир", Москва, 1976.
2. В.В. Митрофанов, Б.В. Одинцов. Программы обслуживания ОС ЕС ЭВМ. Изд-во "Статистика", Москва, 1977.
3. З.С. Брич, В.И. Вокш, Г.С. Дегтярева, Э.В. Ковалевич. Программирование на языке Ассемблера ЕС ЭВМ. Изд-во "Статистика", Москва, 1975.

Рукопись поступила в издательский отдел  
6 декабря 1977 года.