

Б-202



ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

1882 / 2-78

24/IV-78

11 - 11246

В.К.Балашов, В.В.Вицев, Н.В.Горбунов,  
Э.И.Мальцев, Ю.П.Петухов, А.Н.Сухоруков

ВЫВОД ИНФОРМАЦИИ  
ИЗ ПРОГРАММ ПОЛЬЗОВАТЕЛЕЙ  
НА ЭКРАНЫ ДИСПЛЕЕВ ЕС ЭВМ

Дубна 1978

11 - 11246

В.К.Балашов, В.В.Вицев, Н.В.Горбунов,  
Э.И.Мальцев, Ю.П.Петухов, А.Н.Сухоруков

**ВЫВОД ИНФОРМАЦИИ  
ИЗ ПРОГРАММ ПОЛЬЗОВАТЕЛЕЙ  
НА ЭКРАНЫ ДИСПЛЕЕВ ЕС ЭВМ**

**Объединенный институт  
ядерных исследований  
БИБЛИОТЕКА**

Балашов В.К. и др.

11 - 11246

Вывод информации из программ пользователя на экраны дисплеев ЕС ЭВМ

Предлагается описание, текст и инструкции для работы с программой SCREEN, с помощью которой выводится информация из программ пользователей, написанных на языках высокого уровня. Вывод производится на экраны дисплеев произвольного типа, работающих с ЕС ЭВМ в рамках операционной системы ДОС/ЕС любой версии. Предусмотрена возможность сброса содержимого экрана на печать.

Работа выполнена в Серпуховском научно-экспериментальном отделе ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1978

Balashov V.K. et al.

11 - 11246

Data Transmission from User Programs to Display Screen of ES Computers

SCREEN program is described which makes data transmission from programs written on high level programming languages available. The data may be transferred to the display or line-printer. The program works for different types of display under any version of DOS/ES.

The investigation has been performed at the Serpukhov Scientific-Experimental Department, JINR

Communication of the Joint Institute for Nuclear Research. Dubna 1978

## ВВЕДЕНИЕ

Во многих случаях, в особенности при работе с моделирующими и оптимизирующими программами, желательно иметь возможность постоянного визуального наблюдения за ходом изменения значений каких-либо параметров расчета.

Программы, реализующие такую возможность, существуют (см., напр., /1-5/). Однако во многих случаях к ним неудобно обращаться из программ, написанных на языке высокого уровня; кроме того, для работы с ними требуется достаточно глубокое знание конкретных операционных систем. Предлагаемая программа SCREEN проста в обращении, может быть вызвана из программ пользователей, написанных на любом языке, входящем в состав матобеспечения ЕС ЭВМ, и предусматривает знание только лишь размера экрана дисплея (количество символов в строке) и логического номера дисплея.

Программы пользователей могут работать в любом разделе (BG, F2 и F1) всех версий ДОС/ЕС, так как SCREEN каталогизируется в библиотеке перемещаемых модулей.

Вывод информации может быть осуществлен на дисплеи всех типов, работающие с ЕС ЭВМ, (ЕС-7906, ВИДЕОТОН, BSAN и др.) с экранами размером не более 16 строк x 80 символов. Для дисплеев с большими экранами (типа ТЕКТРОНИХ и др.) необходимо изменить в программе две строки, где задается размер буферной области (подчеркнуто в тексте, см. раздел III).

В SCREEN предусмотрена возможность сброса содержимого экрана на печать. Это может быть осуществлено как из программы пользователя, так и с помощью прерывания в любой момент времени с пульта ЕС ЭВМ, что особенно важно для уменьшения малоинформационных архивов.

## I. КАК РАБОТАТЬ С ПРОГРАММОЙ

SCREEN позволяет выводить на экран величины типа REAL, INTEGER (простые переменные и массивы) и любую текстовую информацию. Отметим здесь, что программа работает с данными указанного типа длиной только в одно слово. Кроме того, при использовании ее в языке PL/I в списке вызываемых параметров должны употребляться только переменные.

Обращение из программы пользователя производится следующим образом: CALL SCREEN (N,M,MODE; X), где N,M – номер строки и позиции в строке экрана, с которых начинается вывод, MODE задает тип вывода, а X – выводимая информация. Существует набор простейших инструкций, достаточный для работы в полном объеме. Более сложные инструкции, позволяющие уменьшить число обращений, описаны в I.3.

### I.1. Основные инструкции по работе с программой

```
CALL SCREEN (N,M,1,'XXX...X#')
CALL SCREEN (N,M,2,IX)
CALL SCREEN (N,M,3,AX)
```

При таких обращениях на экран дисплея в строке с номером N, начиная с позиции M, будет выведена помещенная в кавычки фраза, текущее значение величин IX(INTEGER) и AX(REAL) соответственно. Текст в кавычках должен всегда заканчиваться служебным знаком #.

### I.2. Управляющие инструкции

```
CALL SCREEN (0,0,0,0); CALL SCREEN (0,0,6,0); CALL SCREEN(0,K1,7,K2)
```

При этих обращениях происходит очистка экрана, выдача на печатающее устройство копии экрана и конкретизация используемого дисплея соответственно. Если последнее обращение отсутствует, то предполагается K1 = 64, K2 = 20 (K1 – число символов в строке экрана, K2 – логический номер устройства в конкретной операционной системе).

### I.3. Более сложные инструкции

Как видно из предыдущего примера, с помощью простых инструкций можно делать любой вывод на экран, но при этом сразу бросается в глаза наличие большого количества обращений. В этом разделе описываются инструкции, позволяющие объединить в одном вызове несколько команд, ввести кратность обработки и обеспечить удобство работы с массивами.

Наиболее общий вид обращения может быть записан следующим образом:

```
CALL SCREEN (N,M,K1MODE,X1,X2,...,XK1,K2MODE,
'TEXT#,K3MODE,X...)
```

Рассмотрим отдельно каждую часть записи, отметив сразу же, что порядок следования частей произвольный.

K1MODE,X1,X2,...XK1 Коэффициент K1 перед значением MODE указывает число параметров X1, выводимых с одним признаком. Коэффициент ставится перед значением MODE без какого-либо разделяющего символа. Пусть мы выводим величины типа REAL(MODE=3), тогда вызов

```
CALL SCREEN (N,M,23,A1,A2)
эквивалентен двум записям вида
CALL SCREEN (N,M1,3,A1)
CALL SCREEN (N,M2,3,A2)
```

K2MODE, 'TEXT#' задает кратность выводимого текста, что особенно удобно при подчеркиваниях, задании пробелов, повторов и т.п.

Например, вызов (для дисплея с 80 символами в строке)

CALL SCREEN (N,20,1, 'TABLE#',551, '#',801, '-#')  
приведет к следующему: с 20-й позиции строки "N" будет напечатана фраза TABLE, затем последуют 55 пробелов, а на следующей строке будет подчеркивание из 80 символов "-".

Если величина X описана как массив, то обращение

CALL SCREEN (N,M,75,X9)  
выведет на экран первые 7 элементов массива X. (Для массивов типа REAL MODE=5, для типа INTEGER MODE=4).

Отметим здесь, что при комбинированной форме обращения задается только номер строки и позиции начала записи. Дальнейшее расположение информации на экране происходит автоматически по жестким форматам.

#### 1.4. Форматы вывода информации

Форматы вывода являются жестко заданными. Для действительных чисел  $\pm.XXX\pm YY(E8.3)$ , для целых  $\pmXXXXXXXX(18)$ . Таким образом, при выводе только числовой информации в одной строке экрана можно поместить от 8 до 10 чисел в зависимости от типа дисплея. При выводе целых чисел, больше чем 9999999, печатаются символы \*\*\*.

#### 1.5. Сообщения об ошибках обращения

При ошибках в задании MODE,N,M или коэффициента кратности K на экран выводится сообщение SYNTAX? и происходит выход из SCREEN в вызывающую программу.

Остальные ошибки не являются фатальными и приводят к следующим сообщениям на экране:  
NO END#? - в случае отсутствия служебного знака в конце текста. Сам текст не выводится. (При некото-

рых случайных наложениях вместо этой фразы может появиться произвольный набор символов).

INTEGER? - если выводимое действительное число меньше чем  $1.E-78$ , что, как правило, бывает, если целое число передается по формату для действительного числа.  
TOO LONG - если в одном выводе присутствует больше символов, чем может поместиться на экране дисплея.

#### 1.6. Сброс содержимого экрана на печать с пульта ЭВМ

В программе пользователя не всегда можно заранее предусмотреть необходимость вывода информации на печать для получения протокола. Программа SCREEN предоставляет возможность осуществления такого вывода в любой момент времени. При работе программы пользователя в разделах F1 и F2 вывод достигается путем прерывания с пишущей машинки через программу "Внимание" директивой MSG F1(F2), при работе в разделе BG - путем нажатия кнопки "Внешнее прерывание" на пульте оператора. После вывода на печать содержимого экрана программа продолжает работу без каких-либо воздействий со стороны оператора.

## II. КРАТКОЕ ОПИСАНИЕ ОСНОВНЫХ АЛГОРИТМОВ ПРОГРАММЫ

Большая часть программы предназначена для преобразования данных, содержащихся в памяти ЭВМ в символьном, целом и плавающем форматах, в символьное представление. Относящиеся к данной обработке алгоритмы могут быть легко найдены специалистом в приведенном ниже тексте программы.

Отметим здесь лишь два ключевых момента в организации программы. Независимость программы от конкретных устройств вывода обусловлена применением физического уровня программирования. Любые устройства (даже не предусмотренные системой, а включенные при генерации как физические), которые воспринимают

команду записи 01, будут пригодны для выполнения операции вывода. Таким образом, вывод информации можно делать не только на экран дисплея, но также на печатающее устройство, перфоленту и магнитную ленту. Ошибки на устройствах обрабатываются системой стандартным образом. Необходимо однако помнить, что в устройствах с отличной от EBCDIC кодировкой символов (например, ВІДЕОТОН - код ASCII) ) пользователь должен самостоятельно добавить в программу блок перекодировки (команды перекодировки перед выводом на экран и таблицу перекодировки).

Для осуществления вывода на печать содержимого экрана в момент прерывания с пульта оператора в программе предусмотрена связь с супервизором для обработки внешних прерываний. Это накладывает ограничение на установление в программе пользователя собственных связей по обработке прерываний - их наличие автоматически приведет к исключению возможности вывода на печать через пульт оператора.

### III. ТЕКСТ ПРОГРАММЫ SCREEN

```
// EXEC ASSEMBLF
SCREEN  START X'7800'
        PRINT NOGEN
        SAVE (14,12)
        BALR 10,0
        USING **,10
BEG     BC 0,**+18
        LR 2,1
        STXIT OC,INTPRINT,SAVEAREA
        LR 1,2
        OI BEG+1,X'F0'

*
* USING REGISTERS IN PROGRAMM
* REG1 = ADDRES OF PARAMETERS ADDESES AND WORKING REGISTER
* REG2 = CONTAIN 1
* REG3 = ADDRES END OF OUTPUT BUFFER
* REG4 = ADDRES FOR OUTPUT BUFFER
* REG5 = LENGTH OF OUTPUT INFORMATION
* REG6 = ADDRES OF PARAMETERS FOR CONVERSION
* REG7 = WORKING REGISTER 1
* REG8 = WORKING REGISTER 2
* REG9 = LINK REGISTER
```

```
* REG10= BASE REGISTER
* REG11= ADDRES OF PARAMETERS ADDESES
* REG12= COUNTER IN ROUTINES
```

```
* BEGINING ROUTINE
LH 2,=H'1'
L 7,0(1)
BAL 9,TESTMD
L 8,0(7)
SR 7,7
CR 8,7
BE CNTRL
```

```
* COMPUTE BUFFER ADDRES
```

```
SR 8,2
BM MODERR
MH 8,NUMPOS+2
L 7,4(1)
BAL 9,TESTMD
L 7,0(7)
SR 7,2
BM MODERR
AR 7,8
LR 4,7
LA 7,BUFFER
AR 4,7
A 7,LBUF
LR 3,7
SR 5,5
BAL 9,OVERFL
LA 11,8(1)
```

```
CONTIN L 7,0(11)
        BAL 9,TESTMD
        L 7,0(7)
        SR 6,6
        LTR 7,7
        BM MODERR
        D 6,=F'10'
        LTR 12,7
        LR 7,6
        BNZ **+6
        AR 12,2
        SR 7,2
        BE TEXT
        SR 7,2
        BE INTNUM
        SR 7,2
        BE REALNM
        SR 7,2
        BE INTARR
        SR 7,2
        BE REALAR
```

```
*
* ERROR MODE MESSAGE
*
```

```

MODERR MVC BUF,=C' SYNTAX?'
MODER1 LA 4,BUFFER
      A 4,LEBUF
      SH 4,=H'8'
      MVC 0(16,4),BUF
      B EXIT
*
* TEST OVERFLOW BUFFER
*
OVERFL LR 7,4
      AR 7,5
      CR 3,7
      BCR 10,9
      MVC BUF,=C'TOO LONG'
      B MODER1
*
* TEST PARAMETER'S ERROR
*
TESTMD TM 0(7),X'80'
      BCR 8,9
      B MODERR
*
* OUTPUT TEXT ON THE SCREEN
*
TEXT L 6,4(11)
      BAL 9,SYMBOL
      BAL 9,OVERFL
      SR 5,2
      STC 5,#+5
      MVC 0(0,4),BUF
      AR 5,2
      AR 4,5
      BCT 12,TEXT+8
      TM 4(11),X'80'
      LA 11,8(11)
      BZ CONTIN
      B EXIT
*
* INTEGER NUMBERS ONTO THE SCREEN
*
INTNUM L 6,4(11)
      BAL 9,INTEGR
      BAL 9,OVERFL
      SR 5,2
      STC 5,#+5
      MVC 0(0,4),BUF
      AR 5,2
      AR 4,5
      LR 7,11
      BAL 9,TESTMD
      LA 11,4(11)
      BCT 12,INTNUM
      TM 0(11),X'80'
      LA 11,4(11)
      BZ CONTIN
      B EXIT
*

```

```

* INTEGER ARRAY ONTO THE SCREEN
*
INTARR L 6,4(11)
      BAL 9,INTEGR
      BAL 9,OVERFL
      SR 5,2
      STC 5,#+5
      MVC 0(0,4),BUF
      AR 5,2
      AR 4,5
      LA 6,4(6)
      BCT 12,INTARR+4
      TM 4(11),X'80'
      LA 11,8(11)
      BZ CONTIN
      B EXIT
*
* REAL NUMBERS ONTO THE SCREEN
*
REALNM L 6,4(11)
      BAL 9,FLOAT
      BAL 9,OVERFL
      SR 5,2
      STC 5,#+5
      MVC 0(0,4),BUF
      AR 5,2
      AR 4,5
      LR 7,11
      BAL 9,TESTMD
      LA 11,4(11)
      BCT 12,REALNM
      TM 0(11),X'80'
      LA 11,4(11)
      BZ CONTIN
      B EXIT
*
* REAL ARRAY ONTO THE SCREEN
*
REALAR L 6,4(11)
      BAL 9,FLOAT
      BAL 9,OVERFL
      SR 5,2
      STC 5,#+5
      MVC 0(0,4),BUF
      AR 5,2
      AR 4,5
      LA 6,4(6)
      BCT 12,REALAR+4
      TM 4(11),X'80'
      LA 11,8(11)
      BZ CONTIN
      B EXIT

```

```

*
* ROUTINE FOR PRINTING SCREEN FROM DISPLAY UNDER
* INTERRUPTION
*
INTPRINT BALR 11,0
          USING *,11
          BAL 9,PRINT1
          EXIT OC
          DROP 11

*
* CHOICE CONTROL ROUTINE
*
CNTRL    L 7,8(1)
          BAL 9,TESTMD
          CLI 3(7),X'00'
          BE CLEAN
          CLI 3(7),X'06'
          BE PRINT
          CLI 3(7),X'07'
          BE CHANGE
          B MODERR

*
* CHANGE ROUTINE
*
CHANGE   L 7,4(1)
          BAL 9,TESTMD
          L 7,0(7)
          ST 7,NUMPOS
          L 7,12(1)
          L 7,0(7)
          STC 7,A+7
          L 7,NUMLIN
          MH 7,NUMPOS+2
          STH 7,A1+6
          ST 7,LEBUF
          B EXIT

*
* PRINT SCREEN FROM DISPLAY ON SYS1ST
*
PRINT    BAL 9,PRINT1
          B EXIT
* PRINT ROUTINE
PRINT1   BALR 14,0
          USING *,14
          L 7,NUMLIN
          MVC BUF,CLNBUF
          L 8,NUMPOS
          STC 8,**+17
          AH 8,='16'
          STH 8,B1+6
          LA 8,BUFFER
PRINT2   MVC BUF+16,0(8)
          A 8,NUMPOS
          EXCP B

```

```

          WAIT B
          BCT 7,PRINT2
          BCR 15,9
          DROP 14

*
* CLEANING ROUTINE
*
CLEAN    L 5,LEBUF
          LA 6,BUFFER
CLEAN1   MVI 0(6),C' '
          AR 6,2
          BCT 5,CLEAN1
          B EXIT

*
* SYMBOLIC ROUTINE
*
SYMBOL   LR 7,6
          LNR 5,2
SYM1     CLI 0(7),C'#'
          BE SYM2
          AR 5,2
          AR 7,2
          CH 5,=H'255'
          BP SYM3
          B SYM1
SYM2     STC 5,**+5
          MVC BUF,0(6)
          AR 5,2
          BCR 15,9
SYM3     MVC BUF,=C'NO END*?'
          LH 5,=H'8'
          BCR 15,9

*
* INTEGER ROUTINE
*
INTEGR   L 7,0(6)
          CVD 7,DEC
          MVC BUF,CLNBUF
          MVC DECF,DECSH
          EDK DECF,DEC
          BP POSIT
          BZ NUL
          MVI BUF,C'- '

PCSIT    LA 7,DECF+15
          SR 7,1
          CH 7,=H'6'
          BP INTOVF
          STC 7,**+5
          MVC BUF+1,C( 1)
          LH 5,=H'8'
          BCR 15,9
          MVI DECF+15,C'0'
          LA 1,DECF+15
          B POSIT
          MVC BUF(8),=C'*****'
          B POSIT

*
* ROUTINE FOR CONVERSION FLOATING NUMBER IN OUTPUT
* FORMAT E16.6
*

```



```

FLOAT      MVC BUF,NULPNT
           STD 0,FLSAV1
           STD 2,FLSAV2
           L 7,0(6)
           ST 7,FL
           LTR 7,7
           BZ FLEXIT
           BP TSTSMML
           MVI BUF,C'-.'
           NI FL,X'7F'
TSTSMML   CLC FL,SMALL
           BNM L1
           MVC BUF,=C'INTEGER?'
           B FLEXIT
L1         SDR 0,0
           SDR 2,2
           LE 2,D10
           SR 8,8
           LE 0,FL
L2         STE 0,FL
           CLI FL,X'46'
           BZ C1
           BM C2
           AR 8,2
           DDR 0,2
           B L2
C2         SR 8,2
           MDR 0,2
           B L2
* TRANSFORM MANTISSA IN DECIMAL NUMBER
C1         NI FL,X'00'
           L 7,FL
           CVD 7,DEC
* MAKE EQUAL NUMBER DECIMAL DIGITS IN RESULT
C3         CP DEC,MASKP
           BNM L6
           MP DEC,DP10
           SR 8,2
           B C3
L6         AP DEC,DP50000
* SYMBOLIC CONVERSION FOR MANTISSA
           MVC DECF,DECSH
           EDK DECF,DEC
           MVC BUF+2(3),0(1)
* SYMBOLIC CONVERSION FOR EXPONENT
           AH 8,H'8'
           BNM L5
           MVI BUF+5,C'-.'
L5         CVD 8,DEC
           UNPK DECF(3),DEC+6(2)
           OI DECF+2,X'F0'
           MVC BUF+6(2),DECF+1
           LD 0,FLSAV1
           LD 2,FLSAV2
FLEXIT     LH 5,=H'8'
           BCR 15,9
*
* OUTPUT ROUTINE
*

```

```

OUTPUT     EXCP A
           BCR 15,9
*
* RETURN TO CALLING PROGRAMM
*
EXIT       BAL 9,OUTPUT
           RETURN(14,12)
           CNOP 0,8
A          CCB SYS020,A1
A1         CCW 1,BUFFER,0,1024
B          CCB SYSLST,B1
B1         CCW 9,BUF,0,80
FLSAV1    DC D'0'
FLSAV2    DC D'0'
DEC        DC XL8'0'
DECF       DC CL16' '
DECSH      DC C' '
           DC 15X'20'
SMALL      DC E'1.E-78'
NUMPOS     DC F'64'
NUMLIN     DC F'16'
LBUF       DC F'1024'
D10        DC E'10'
FL          DC E'0'
SAVEAREA   DC 72X'0'
NULPNT     DC C' .000+00'
MASKP      DC X'9995000C'
DP10       DC X'010C'
DP50000    DC X'50000C'
CLNBUF     DC C' '
BUF         DC 32CL8' '
BUFFER     DC 1280C' '
           END
/*

```

В заключение мы хотим поблагодарить М.И.Соловьева и Н.Н.Говоруна за поддержку наших работ по математическому обеспечению, Л.П.Холоденко и Х.Эшер за большую техническую помощь в подготовке данной программы.

#### ЛИТЕРАТУРА

1. Неустроев П.В., Семенихин В.И. Препринт ЛИЯФ, №166, 1975.

2. Аульченко В.М., Коршунов Ю.В., Мелихов Г.Г.  
Препринт ИЯФ СОАН СССР, 76-1, 1976.
3. Кузнецов В.Н. ОИЯИ, 10-8994, Дубна, 1975.
4. Ефимов Л.Г., Пискунов Н.М., Ситник И.М. ОИЯИ,  
10-8833, Дубна, 1975.
5. Смирнов В.А. ОИЯИ, 10-9302, Дубна, 1975.

Рукопись поступила в издательский отдел  
23 января 1978 года.