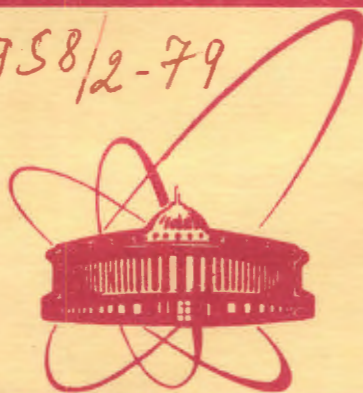


4958/2-79



сообщения
Объединенного
Института
Ядерных
Исследований
Дубна

Ц8405
Д-691

3/12-79

10 - 12480

В.И. Дорогов, М. Кунике

ПОДПРОГРАММЫ ДЛЯ ДОСТУПА
К НЕКОТОРЫМ СИСТЕМНЫМ КОМПОНЕНТАМ ОС ЕС
ПРИ ПРОГРАММИРОВАНИИ НА ЯЗЫКЕ ФОРТРАН

(II)

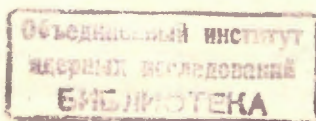
1979

10 - 12480

В.И.Дорогов, М.Кунике

ПОДПРОГРАММЫ ДЛЯ ДОСТУПА
К НЕКОТОРЫМ СИСТЕМНЫМ КОМПОНЕНТАМ ОС ЕС
ПРИ ПРОГРАММИРОВАНИИ НА ЯЗЫКЕ ФОРТРАН

(II)



Дорогов В.И., Кунике М.

10 - 12480

Подпрограммы для доступа к некоторым системным компонентам ОС ЕС при программировании на языке ФОРТРАН

Предлагаются описания и примеры использования фортрановских подпрограмм СНКРТ /для создания контрольной точки и повторного запуска программы с контрольной точки/, TIME /для установления даты и времени дня/ и группы подпрограмм STIMER, TTIMER, CTIMER /для работы с таймером/.

Подпрограммы написаны на языке АССЕМБЛЕР и работают под управлением системы ОС ЕС любой версии с таймером.

Работа выполнена в Лаборатории высоких энергий ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1979

Dorogov V.I., Kunicke M.

10-12480

Subroutines for Using Some System Components of OS ES In Programming on FORTRAN

Description and examples of the use of CHKTP subroutines (to set a checkpoint and restart of the program from a checkpoint), TIME (to get data and time) and group of subroutines STIMER, TTIMER, CTIMER (for set, test and cancel the times) are given. Subroutines are written in ASSEMBLER language and are applied in the OS/ES operation system with a timer.

The investigation has been performed at the Laboratory of High Energies, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1979

Настоящее сообщение является продолжением работы /1/.

Предлагается описание фортрановских подпрограмм, служащих для организации контрольных точек, работы с таймером и установления даты и времени дня.

Для всех программ приведены примеры использования. При этом предполагается, что подпрограммы находятся в библиотеке загрузочных модулей с именем USR1.PGMLIB.

ФОРТРАНОВСКАЯ ПОДПРОГРАММА СНКРТ/КОНТРОЛЬНАЯ ТОЧКА/

Аномальное окончание работы программы обычно приводит к полной потере времени счета или даже всей информации /последнее, например, имеет место в случае сбора данных/. Полной потери информации и времени можно избежать, если использовать имеющуюся системную компоненту "контрольная точка/рестарт". Она позволяет возобновить обработку информации программой не с самого начала, а с определенного момента, с так называемой контрольной точки.

Подпрограмма СНКРТ дает программисту, использующему язык ФОРТРАН-4, возможность обращаться к системной компоненте "контрольная точка/рестарт".

Структура:

подпрограмма

имя подпрограммы: СНКРТ

вызываемых программ нет

вызов CALL СНКРТ(< идентификатор >, < код возврата >)

параметры блока управления набора данных:

DSORG=PS, MACRF=(W), RECFM=U,

DDNAME=FTCHKPT

язык программирования: АССЕМБЛЕР

операционная система ОС/ЕС.

Использование:

Подпрограмма СНКРТ вызывается фортрановским оператором CALL. При выполнении подпрограммы в набор данных FTCHKPT записывается контрольная точка.

Идентификатор контрольной точки указывается в первом параметре. В случае, если до вызова CHKPT был запущен счетчик времени для данной задачи при помощи вызова подпрограммы CALL STIMER, контрольная точка не записывается. Для того, чтобы в этом случае иметь возможность записать контрольную точку, необходимо остановить счетчик времени вызовом CALL STIMER. Для возобновления работы программы можно запросить автоматический или отсроченный рестарт /повторный запуск/ с контрольной точки /метод запроса - смотри ниже/.

Параметры подпрограммы. Первый параметр задает идентификатор записываемой контрольной точки. Его можно задать либо в виде переменной длиной 8 байтов, которой с помощью оператора DATA присвоена в качестве значения холлерическая константа, либо непосредственно в виде холлерической константы длиной в 8 символов. Вторым параметром является выходным, он должен быть описан как INTEGER*4. Если его значение равно 0, то это говорит о том, что была записана контрольная точка, если оно равно 4, то производился рестарт с данной контрольной точки, если же это значение больше 4, то, следовательно, при записи контрольной точки произошла ошибка, т.е. этой контрольной точкой нельзя воспользоваться для рестарта.

Набор данных FTCHKPT является последовательным набором данных (DSORG=PS) и состоит из записей неопределенной длины (RECFM=U). В операторе DD с DDNAME=FTCHKPT пользователь должен определять размещение этого набора на внешнем носителе, указывая параметр UNIT, набор может находиться либо на магнитной ленте, либо на сменном диске. Для идентификации набора и, если желательнее, для каталогизации его необходимо указать параметр DSNAME.

Если набор размещается на сменном диске, то следует резервировать соответствующее число дорожек для записи контрольных точек. Необходимое число дорожек в основном определяется размером раздела памяти, в котором выполняется данное задание, и числом наборов данных для данного шага. Число дорожек для одной контрольной точки /если нет поколений наборов данных и нет наборов, занимающих более 5 носителей/ примерно равно $N = M/14 + A/3,6 + 2$, при этом M - число файлов, а A - размер раздела в килобайтах. В зависимости от числа контрольных точек в наборе следует указать подпараметры в параметре SPACE=(TRK, n).

Параметр нормальной диспозиции управляет последовательностью записи контрольных точек в набор: если DISP=NEW или DISP=OLD, то контрольная точка пишется как един-

ственная в начале набора, если же DISP=MOD, то текущая контрольная точка записывается за предыдущими контрольными точками набора.

МЕТОД ЗАПРОСА ЗАПИСИ КОНТРОЛЬНОЙ ТОЧКИ И РЕСТАРТА С КОНТРОЛЬНОЙ ТОЧКИ

Запросы записи контрольной точки и автоматического рестарта осуществляются при помощи RD-параметра в EXEC- или JOB -карте. Отсроченный рестарт заказывается параметром RESTART в JOB-карте. Параметр RD позволяет отменить запись контрольной точки и заказывать автоматический рестарт. Возможны следующие значения параметра:

- RD = R - после записи контрольной точки при сбое работы программы запрашивается автоматический рестарт с контрольной точки.
- RD = NR - отменяется автоматический рестарт, контрольная точка записывается, она может быть использована для отсроченного рестарта.
- RD = NC - отменяется и автоматический рестарт, и запись контрольной точки.
- RD = RNC - запрос автоматического рестарта сохраняется, нет записи контрольной точки и выдачи нового запроса автоматического рестарта.

Параметром RESTART в JOB-карте заказывается отсроченный рестарт с контрольной точки. Он имеет следующий вид:

RESRART = (< имя шага > , < идентификатор контрольной точки >).

В первом подпараметре указывается шаг, на котором контрольная точка была записана. Если этот шаг является шагом каталогизированной процедуры, то параметр имеет вид: < имя шага > . < имя шага процедуры >. Во втором параметре указывается идентификатор контрольной точки, который должен совпадать с параметром вызова CHKPT для соответствующей контрольной точки.

Для осуществления отсроченного рестарта в задании в JOB-карте нужно указать RESTART-параметр, а перед первой EXEC-картой должна находиться DD-карта с DD-именем SYSCHK. Она описывает набор с контрольными точками. Ее параметры должны соответствовать параметрам набора FTCHKPT. Подпараметр диспозиции должен иметь значение DISP = OLD

ПРИМЕР ИСПОЛЬЗОВАНИЯ ПОДПРОГРАММЫ CHKPT

Предполагается, что набор для контрольной точки каталогизирован в системном каталоге и имеет название DSNNAME=USERFILE.

В первом примерном задании при каждом сотом цикле работы программы записывается контрольная точка, автоматический рестарт при сбое работы программы отменён. Контрольная точка записывается всегда в начале набора, т.е. сохраняется только последняя контрольная точка. Во втором задании производится отсроченный рестарт с контрольной точки, запись контрольной точки отменяется.

```
// EXAMPLE1 JOB HUSER, PROGRAMMER, CLASS=A,
//          MSGLEVEL=1,RD=NR
// P1      EXEC FORSTCLD
// FORT.SYSIN DD *
INTEGER*4  CHCODE
REAL*8     CHNAME
DATA CHNAME /'CHECKPNT' /
I=0
100 I=I+1
IF(I.LT.100) GOTO 200
I=0
CALL CHKPT (CHNAME, CHCODE)
IF (CHCODE.EQ.0) GOTO 200
IF (CHCODE.EQ.4) GOTO 300
IF (CHCODE.GT.4) GOTO 400
300 PRINT 301
200 CONTINUE
.
.
GOTO 100
400 PRINT 401
.
.
301 FORMAT (10X, 'RESTART FROM CHECKPOINT')
401 FORMAT (10X, 'ERROR DURING CHECKPOINT - WRITING')
END
```

/*
6

```
// GO.SYSLIB DD
//          DD DSN=USR1.PGMLIB, DISP=SHR
// GO.FTCHKPT DD DSN=USERFILE, DISP=OLD
//
// EXAMPLE2 JOB HUSER, PROGRAMMER, MSGLEVEL=1,
//          RESTART=(P1.GO, CHECKPNT), RD=NC
// SYSCHK   DD DSN=USERFILE, DISP=OLD
// P1      EXEC FORSTCLD
// FORT.SYSIN DD DUMMY
// GO.SYSLIB DD
//          DD DSN=USR1.PGMLIB, DISP=SHR
// GO.FTCHKPT DD DUMMY
//
```

Замечание. В задание для отсроченного рестарта должны входить те же DD-карты, что и в задание, в котором была записана контрольная точка. Параметры их могут быть изменены. Кроме того, можно добавить DD-карты /например, SYSUDUMP/.

JOB -карта задания, в котором записывается контрольная точка, и JOB -карта задания, в котором выполняется отсроченный рестарт, должны содержать параметр MSGLEVEL=1

ФОРТРАНОВСКАЯ ПОДПРОГРАММА УСТАНОВЛЕНИЯ ДАТЫ И ВРЕМЕНИ ДНЯ /TIME/

Для идентификации результатов счета, полученных в разное время, а также для организации различных ветвей обработки по временному признаку удобно использовать идентификатор, соответствующий дате или времени дня. В языке ФОРТРАН-4 отсутствуют стандартные программы для получения этих характеристик.

Подпрограмма TIME предназначена для установления даты и времени дня.

Структура подпрограммы TIME

Подпрограмма

имя подпрограммы: TIME

вызываемых подпрограмм нет

вызов подпрограммы: CALL TIME (< пар 1 >, < пар 2 >)

язык программирования: АССЕМБЛЕР

операционная система ОС/ЕС.

Использование подпрограммы TIME

Подпрограмму TIME можно вызывать фортрановским оператором CALL. Параметрами подпрограммы должны быть переменные длиной 8 байтов.

Параметры подпрограммы. Параметр <пар 1> задает область памяти, в которой будет размещаться текущая дата в форме DD /MM/ YY, где

DD - день месяца,
MM - порядковый номер месяца,
YY - последние две цифры года.

Параметр <пар 2> задает адрес памяти, по которому размещается текущее время дня в форме HH.MM.SS, где

HH - часы,
MM - минуты,
SS - секунды.

Алгоритм работы подпрограммы. Работа подпрограммы основана на использовании макрокоманд супервизора TIME и преобразовании информации, полученной от супервизора, в удобную форму. Необходимо отметить, что точность полученной информации зависит от точности задания параметров DATA и CLOCK в команде оператора SET.

Пример вызова подпрограммы

```
//EXAMPLE      JOB      HUSER, NAME
//FORTRAN      EXEC      FORTGCL6
//FORT.SYSIN   DD          *
                LOGICAL*1  DATA(8), CLOCK(8)
                CALL TIME   (DATA, CLOCK)
                WRITE (6,1)  DATA
                WRITE (6,1)  CLOCK
1 FORMAT (1X, 8A1)
                STOP
                END

/*
//LIB SYSLIB   DD
//          DD DSN=USR1.PCMLIB, DISP=SHR
//
```

ФОРТРАНОВСКИЕ ПОДПРОГРАММЫ ДЛЯ ОБРАЩЕНИЯ К ТАЙМЕРУ/STIMER, TTIMER, CTIMER/

Для организации различных ветвей обработки в зависимости от времени счета необходимо иметь доступ к таймеру.

Подпрограмма STIMER запускает таймер данной задачи, подпрограмма TTIMER устанавливает время процессора, затраченное после запуска таймера этой задачи, подпрограмма CTIMER останавливает таймер.

Структура подпрограммы STIMER

Подпрограмма

имя подпрограммы: STIMER

внутренние входы: TTIMER, CTIMER

вызываемых подпрограмм нет

вызов: 1/ CALL STIMER - запустить таймер

2/ CALL TTIMER (<пар 1 >) - опросить таймер

3/ CALL CTIMER - остановить таймер.

Язык программирования: АССЕМБЛЕР

Операционная система: ОС/ЕС

Использование подпрограмм STIMER, TTIMER, CTIMER

Подпрограмму STIMER можно вызывать фортрановским оператором CALL. При обращении к подпрограмме STIMER происходит установление таймера на определенное время, приблизительно 7,5 часа. К подпрограмме STIMER можно обращаться и повторно в любом месте программы. Тогда оставшийся интервал времени будет заменен начальной величиной 7,5 часа.

Для обращения к подпрограмме TTIMER можно пользоваться фортрановским оператором CALL с одним параметром типа INTEGER*4.

Для обращения к подпрограмме CTIMER можно пользоваться фортрановским оператором CALL, в результате чего таймер будет остановлен.

Параметр подпрограммы TTIMER. Параметр <пар 1> задает область памяти, в которой будет размещаться число, показывающее время процессора, прошедшее после запуска таймера. Время задается в сотых долях секунды как целое число.

Алгоритм работы подпрограмм. Работа рассмотренных выше подпрограмм основана на использовании макрокоманд супервизора STIMER и TTIMER и соответствующих преобразований к удобной форме. Точность полученной информации - порядка

сотых долей секунды. Она зависит от точности работы аппаратного таймера.

Пример вызова подпрограммы.

```
//EXAMPLE JOB HUSER, NAME
//FORTRAN EXEC FORTGCLD
//FORT.SYSIN DD *
      INTEGER*4 BEGTIM
      CALL STIMER
      .
      .
      .
      .
      CALL TTIMER (BEGTIM)
      PRINT 1, BEGTIM
      .
      .
      .
      CALL CTIMER
      1 FORMAT (1X, I8)
      STOP
      END

/*
//LIB.SYSLIB DD
// DD DSN=USR1.PCMLIB, DISP=SHR
//
```

ЛИТЕРАТУРА

1. Дорогов В.И., Кунике М. ОИЯИ, 10-12278, Дубна, 1979.

В библиотеке STENLIB есть процедура TIMEI (T)

вещ. в секундах с точностью до сотых,

Рукопись поступила в издательский отдел
23 мая 1979 года.

и TIMEI (T) — тоже первый вид, определяет время промешу. время, а секунды и время от начала промешу.