Note on Cosmos version 6.35 or newer

1 The file name convention

In older versions, if a file name given in the namelist data had `%_` the `%` was replaced by the hostname where the program ran. The `#_` was also treated specially. This convention is changed as follows from this version so that the unique file name can easily be set up on multi-cpu machines.

Table 1: Special file name ingredient

special symbols in a file name	every `@`, `#` or `%` is replaced by (if AtEnv etc is ' ')	related 2nd group namelist variable
`@`	hostname	AtEnv
`#`	unix process number	SharpEnv
`%`	YYMMDDHHMMSS	PercentEnv

The table tells that if the namelist variable shown above, say, *AtEnv*, is blank (**default**), every `@` in a file name given in the namelist data (say, by ContFile=...), will be replaced by the host name where the program will run. In some case, host name may be expressed with the complete domain name. In such a case, the domain name will be dropped. If *AtEnv* is non blank, then the system tries to get the value of the environmental variable expressed by the string value of *AtEnv*, and the value is used to replace `@`. If it cannot be obtained, the system will stop after issuing a message. `#` and `%` are treated similarly. YYMMDDHHMMSS is the 12 character string like 030425102305 which means Year 2003 April 25th 10 O'clock 23 min 05 sec.

For example, if ContFile = `'../@/#_Continfo'`, it will become `'../xyz/1234_Continfo'` for the host 'xyz' and process number '1234'.

2 Some new routines

The user may use the following routines.

- *kgetenv2* to get value of an environmental variable.

  i = kgetenv(env, envvalue)

  where

```
integer i              ! output. if 0, envvalue is obtained. else
                       !         env not exist.
character*n  env       ! input   string shown by 'env' is an
                       !         environmental variable
                       !         n must be the appropriate length
character*m  envvalue  ! output  value of the environmental
                       !         variable.
                       !         m must be the appropriate length
```

- *kgettime* to get time in sec as explained in man (3) time.

  i = kgettime(dummy)

  where

```
      integer i          !  output.  time value
integer dummy     !  input.   not used.
```

- *kgetnow* to get character string representation of date up to second.

  i = kgetnow(yymmdd)

  where

```
       integer i  !  output.   12 which is the length of yymmdd
       character*n  yymmdd  output.   n >= 12.  YYMMDDHHMMSS.
```

- *kgetpid* to get the unix process number of the currently running program.

  i = kgetpid(dummy)

  where

```
      integer i   !  output.  obtained process number.
      integer dummy ! input.  not used.
```

- *cgetfname* to convert file name as explained earlier (`@,#, and`or

  call cgetfname(fin, fout)

  where

```
       character*n fin   !  input file name which may has \verb/@# or %/
       character*m fout  !  outupt file name with \verb/@# or %/ replaced.
```

3 Random number seed

If both of InitRN(1) and InitRN(2) are positive, they are used as the seed for all the random numbers in the run. This is the same as in the all versions. Also as in earlier versions, if InitRN(2) is < 0, the random number seed is generated in Cosmos. However, the the method is modified; |InitRN(2)|, unix process number, timer value and host name are used for the generation. If InitRN(1) < 0, the treatment is the same as the earlier versions. That is, the seed of each event is obtained by reading a file opened by the user with fortran logical file number 14. The file should contain 2 seeds in each line.