```
/* formate --: formats the current or next paragraph

   A paragraph is defined by consecutive text separated by empty lines.
   The macro formats the whole paragraph, irrespective the position of
   cursor within the paragraph. The paragraph indentation is not recognized.
   The macro removes also subsequent spaces. The text is broken at word
   ends. Word delimiters are always REXX's word delimiters, not THE
   delimiters (THE can toggle the way words are delimited).

 */
/* Date: Mon, 26 Dec 1994 22:23:46 +0100 (MEZ) */
/* Klaus H. Ringhofer */
/* Department of Physics */
/* University */
/* D - 49069 Osnabrueck */
/* GERMANY */
/*  */
/* Phone:   + 49 969 541 2685 */
/* Fax:     + 49 541 969 2670 */
/* e-mail:    ringhofe@physik.uni-osnabrueck.de */

/* donated to the public without any warranty of any kind */

/*
   adapted to THE 2.6 by Franz-Josef Wirtz, fw@ecotopia.oche.de
   (12 October 1998)
*/

/* known bugs:
   * When first line is longer than margins, a space occurs in front of
     first line.
   * When first line is longer than two lines the second line doesn't
     flow.

  Limitations: didn't work on hpux until THE 1.5, maybe it does now...
*/

/* Exclude: cursor on commandline or end of file */
IF incommand() THEN DO
  'emsg Cursor should be in text area to format paragraph'; EXIT
END
IF eof() THEN DO
  'emsg Cursor already on end of file, nothing to format'; EXIT
END

/* TRACE ?results */

/* Get current margins; parindent not taken into account: I use TeX */
'extract /margins/'
left_col = margins.1
right_col = margins.2
/*
   left_col = 1
   right_col = 72
*/
text_width = right_col - left_col + 1

/*We go to the beginning of the next or current paragraph */
'extract /curline/'
IF (curline.3 = '') THEN DO /* go to next paragraph or end of file */
  DO WHILE ((curline.3 = '') & \eof())
```

```
    'down';
    'extract /curline/'
  END
  IF eof() THEN DO
    'emsg No more paragraphs in this file to format'; EXIT
  END
END
ELSE DO /* go up to blank line or top of file */
  DO WHILE ((curline.3 \= '') & \tof())
    'up'
    'extract /curline/'
  END
  'down'
END

/* We are on the first line of a paragraph */
rest='' /* Initiate variables */
'extract /curline/'
this_line=SPACE( curline.3)
DO WHILE ((this_line \= '') & \eof()) /* Process paragraph line for line */
  CALL formate_process_this_line
  'extract /curline/'
  this_line = SPACE( curline.3)
END
IF (rest \= '') THEN DO
  IF (eof()) THEN DO
    'up'
    'sos addline'
    'replace' rest
    'down'
  END
  ELSE DO
    'replace' rest
    'sos addline'
  END
END
EXIT

/* subroutine ---------- */
formate_process_this_line:
   IF (LENGTH( rest) + LENGTH( this_line) + 1 <= text_width) THEN DO
      IF (rest \= "") THEN
         rest = rest' 'this_line;
      ELSE
         rest = this_line
      'sos linedel'
   END
   ELSE DO
      PARSE VAR this_line next_word this_line
      DO WHILE (LENGTH( rest) + LENGTH( next_word) + 1 <= text_width)
         rest = rest' 'next_word
         PARSE VAR this_line next_word this_line
      END
      'replace' rest
      rest = next_word' 'this_line
      'down'
   END
RETURN
```