RCSL No: 31-D610

Edition: October 1981

Author: John Munkholm Andersen
Edith Rosenberg

Title:

Parent Messages in RC8000

RCSL 42-i1592

A/S REGNECENTRALEN af 1979

Keywords:

RC8000, operating system, s, BOSS, parent message, manual.

Abstract:

This manual describes the parent messages used in the RC8000 Multiprogramming System for communication between a job and the Operating System. It also describes how the operating systems 's' and 'BOSS' handle the messages.

(52 printed pages)

Printed by A/S Regnecentralen af 1979, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

RCSL 42-i1592

TABLE OF CONTENTS PAGE

APPENDIX:

1. INTRODUCTION

In a multiprogramming environment there has to be an operating system to control (or at least facilitate) the sharing of resources between jobs. This again calls for the definition of a language for communication between the jobs and the operating system.

In the RC8000 Multiprogramming System (see ref. [11]), a common language has been defined for all operating systems. It is implemented by means of messages sent from the jobs (internal processes) to the operating system (the parent).

Ref. [11] contains an introduction to the concept of messages as a means of communication.

This manual describes the parent messages used. Chapter 2 contains general information, and chapter 3 describes each of the parent messages separately.

Please notice that the section numbering in chapter 3 follows the numbering of the parent message functions (see 2.1), i.e. only even numbers occur.

2. GENERAL INFORMATION 2.

2.1 Message Format 2.1

All parent messages consist of 8 words, like this:

m+0:	function < 12 + pattern < 5 + wait
m+2:	integer or text portion
m+4:	integer or text portion
... until m+14	

function specifies the operation to be performed. Only even values are allowed.

pattern specifies how the message is to be displayed to the operator. The pattern contains seven bits, one to each of the words m+2 to m+14. A bit being one means that the corresponding word contains an integer, otherwise the word contains a text portion of 3 characters (3 NULs denote a blind word). Thus, e.g., bit 1 < 11 means that m+2 is an integer, 1 < 5 that m+14 is an integer.

wait may be zero or one. A zero means that the job wants the answer immediately, a one means that the job wants the answer when the operation is complete, (or when the operator answers the message displayed.)

The notation used will be familiar to assembler programmers. ALGOL programmers will need to know that m+0 is the first word of the message, m+2 the second etc., and that, e.g. 1<5 means: 1 shift 5. Ref. [5] 2.149 (system, example 5) contains an instructive example.

2.2 Answer Format 2.2

The answer to a parent message is also 8 words but the format depends on the operating system as well as on the message:

The operating system "s" unconditionally delivers an answer containing an exact copy of the message sent.

The operating system system BOSS gives an answer as described for each message in chapter 3. Where nothing is stated, the first 3 words of the answer will contain zeroes, and the last 5 words are undefined.

2.3 Operating System Actions 2.3

The Operating System s: The following description is adapted from ref. [10] 2.2:

When the operating system s receives a message from one of its childs, the following is done:

a) If wait (see 2.1) equals 1, the child process is stopped, an answer (see 2.2) is sent to the process and a message is printed on the console, from which the process was created:

```
pause <processname><contents of message>
```

The operator can now start, break or remove the process, depending on the function in the message.

b) If wait equals zero, the answer is sent to the process, and a message is printed on the console from which the process was created:

```
message <processname><contents of message>
```

These are the only actions taken for all parent messages except 'finis' and 'replace', see 3.2 and 3.10.

The Operating System BOSS: The actions of the operating system BOSS are described separately for each parent message in chapter 3, but some common principles may be stated here:

Undefined parent messages are treated as described in 3.16.

The guiding texts, like <:finis:> etc. and actual value of <pattern> (cf. 2.1) are ignored, except for 3.16, where they are used to format the message displayed to the operator (very much like the message displayed by s (see above).

BOSS always uses the main console as display, in contrast to s, which uses the console from which the job was started.

BOSS often ignores the actual value of wait, but the values shown should be used anyway. They have been carefully chosen to ensure that a job may be executed under different operating systems with as little change as possible.

2.4 How to Send Parent Message 2.4

Most user's needs for sending parent messages are covered by the IMPLITIC USE and the dedicated 'job control' UTILITY PROGRAMS mentioned separately for each parent message in chapter 3.

If you have more specialized needs you may send any parent message from a high-level program by means of the standard procedure 'system', see ref. [5] 2.149, where example 5 contains an instructive example.

3.2 Finis 3.2

```
m+0        2<12 + 16<5 + 1
 +2 ┐
 +4 ┘      <:finis_:>
 +6        <param>
```

The job reports that it has finished in the normal way. The operating system will terminate the job.

<param> is a bit pattern specifying special actions. At present the following values are used:

<param> extract 1 = 0: print primary output
<param> extract 1 = 1: do not print primary output.

UTILITY PROGRAM: finis (see ref. [7]) sends the message with wait = 1 and pattern = 0. <param> is extended with the text <:fp:> (left justified).

OPERATING SYSTEM ACTIONS:
s (only MONITOR release 7.0 and later): if WAIT=1 the job process will be removed. See ref. [10] 2.2.

BOSS: terminates the job like this:
- remove links used by the job,
- remove job process (cf. ref. [12] 2.64),
- terminate the use of jobcontrolled printer,
- clean catalog (i.e. remove temporary entries left over by the job),
- terminate the use of jobcontrolled cardreader and tapereader,
- release magnetic tapes and stations used by the job; make scratch tapes public,
- release operator requests concerning the job,
- release unused resources,
- maybe initiate the printing of primary output,
- perform accounting for the job,
- maybe start a 'replace'-job.

3.4 Break

```
m+0    4<12 + 0<5 + 0
 +2 ┐
    } <:break_:>
 +4 ┘
```

The job asks the operating system to perform the "dump and restart"-action (see below).

UTILITY PROGRAM: None.

IMPLICIT USE: This message is sent by all utility programs and by all programs written in high-level languages as the last action taken in response to an internal interrupt (a 'break'). High-level language programs running without fp terminate by sending this message, containing the end program text or an alarm text, see ref. [4] 10.3.2.2 and 10.3.6.2.

OPERATING SYSTEM ACTIONS:
BOSS handles the message, according to the previous history of the job, as follows:

1) The interrupt was a parent break (break cause = 8) forced by BOSS itself in order to abort the job, e.g. as an error reaction to one of the other parent messages, or because the job had exceeded its run time, or because the job was killed by the user or the operator.

2) The interrupt was a parent break (break cause = 8) forced by BOSS but at the explicit request of the job (by means of the parent message 3.28 "timer").

3) The interrupt was not a parent break.

4) No interrupt occurred at all.

In case (1) BOSS will terminate the job as described in subsection 3.2 (above).

In case (2), (3) and (4) BOSS will perform the "dump and restart"-action:

The job process is stopped. If a backing storage file by the name <:image:> is visible from the stdbase of the job and is within the maxbase of the job, the job process will be dumped in the file, and 'contents' in the file descriptor will be changed to 7 ("dumped process"). Then the job process will be removed and created again with claims and user rights (to peripherals) exactly as before the removal. Next, <:fp:> is loaded (from system base) and initalized to continue reading/writing at the current position of the jobfile/primary output file.

FURTHER INFORMATION: Internal interrupts are described in ref. [11] 8.1 and in ref. [12] 1.2.

The actions of the standard interrupt handling routine are described in ref. [6] 3.5 (together with an explanation of the break causes possible), and in ref. [8] 4.2.

In ALGOL (release 8.0 and later releases) you may program your own non-standard interrupt handling actions by means of the standard identifiers 'alarmcause' and 'trapmode' and the standard procedure 'trap'. This is described in ref. [5] 2.5, 2.153 (especially notice example 2) and 2.154.

3.6 Hard error

m+0	6<12 + 16<5 + 0
+2	<:status:>
+4	
+6	<logical status word>
+8	<name of device or area>
+10	
+12	
+14	

The job reports that it has detected a hard error on a peripheral device or a backing storage area.

ANSWER: See 3.16.

UTILITY PROGRAM: None

IMPLICIT USE: This message is sent by all utility programs if they terminate because of hard error on a file. See ref. [6] 6.4 and 4.5.

OPERATING SYSTEM ACTIONS: Handled as 3.16.

FURTHER INFORMATION: The format of the <logical status word> is described in ref. [6] 6.4.

3.8 Account

m+0	8<12 + 15<5 + 0
+2	<:account_:>
+4	
+6	
+8	<param 3>
+10	<param 4>
+12	<param 1> (account kind, must be > = 100)
+14	<param 2>

The job wants to generate a special record on the system account file.

UTILITY PROGRAM: account (ref. [7]). The numbering of the parameters follows the natural numbering of the parameters for the utility program 'account'.

PREREQUISITES:
BOSS: Each use of this message costs an 'account', which must be ordered when the job is started; see ref. [3] 3.3.

ERROR REACTIONS:
BOSS: If the job has no more accounts left, the job will be aborted.

FURTHER INFORMATION: ref. [1] chapter 6 is a detailed description of the accounting system in BOSS2.

3.10 Replace

m+0	10<12 + 16<5 + 1
+2	<:finis_:>
+4	
+6	<param>
+8	<name of replacement job file>
+10	
+12	
+14	

The job specifies a replacement job.

ANSWER (BOSS): The value of the first word of the answer signals the result:

0: ok

1: not ok (replace not accepted from online job)

UTILITY PROGRAM: replace (ref. [7]).

PREREQUISITES (BOSS): The file containing the replacement job must be permanent. The sending job must not be an online job (go or run).

OPERATING SYSTEM ACTIONS:

s (only MONITOR release 7.0 and later): if WAIT = 1, s will remove the sending process immediately and start reading from the replacement job file (which is thus supposed to contain s-commands). See further ref. [10] 2.2 and the 'read'-command, ref. [10] 1.6.

BOSS: The information from the message is stored to be used when the sending job eventually finishes. A replace message cancels previously sent replace messages. The existence and the contents of the replace job file is not checked before the replacement is actually carried out. The replacement will be carried out even if the sending job is aborted (e.g. because of 'time exceeded), except if the job is killed by the operator.

If <param> = -1 (all ones) the message cancels any previously sent replace-message (without specifying a new replacement job).

Otherwise <param> is treated as a bitpattern with the meaning:

<param> extract 1 = 0: take username, userindex and project number from the sending job.

<param> extract 1 = 1: take username, userindex and project number from the replacement job file.

ERROR REACTIONS: see ANSWER.

FURTHER INFORMATION: Notice the difference between how s and BOSS handle the message.

3.12 New job 3.12

```
m+0      12<12 + 0<5 + 0
 +2      <:job:>
 +4  ⎫
 +6  ⎭   <printerspec>
 +8  ⎫
 +10 ⎪
 +12 ⎬   <name of the job file>
 +14 ⎭
```

The job specifies a jobfile to be enrolled for execution.

ANSWER (BOSS): the first word of the answer signals the result:
 0: ok (job file accepted, new job enrolled)
 1: job queue full
 2: file not permanent
 3: file does not exist
 5: file unreadable
 7: user index too large
 8: illegal identification
 9: user index conflict
11: file too long
19: attention status at remote terminal
20: device unknown
21: device not printer
22: parent device disconnected
38: option unknown
39: param at job
40: syntax at job
41: line too long

UTILITY PROGRAM: newjob (ref. [7]), bossjob (ref. [7]), remotejob (ref. [13]). Concerning bossjob and remotejob, see FURTHER INFORMATION (below).

PREREQUISITES (BOSS): The job file of the new job must be permanent, and it must be visible from the standard base of the sending job.

OPERATING SYSTEM ACTIONS:
BOSS: The job file, the printer, and the job specification in the job file is checked and the result is signalled in the answer. If the result is 'ok', the new job is enrolled for execution and will continue independently of (and maybe in parallel with) the sending job.

<printerspec> specifies where primary output from the job is to be printed, and is used as default value for printer specification in converts from the new job. The interpretation is:

0, 0: printer specification is taken from the job which sends the newjob message. If this job is a process running in parallel with Boss, printerspec = 0, 0 works as <:std:>, 0 (see below).

<:std:>, 0: standard printer is used.

anything else: <printerspec> is taken to be the name of a file descriptor or the name of a device in a device controller, and is interpreted as shown in ref. [3], section 7.2. The text in <printerspec> need not be terminated by the usual NULL-character, so a maximum length of 6 characters is possible.

ERROR REACTIONS: see ANSWER (above).

FURTHER INFORMATION: This message may not only be used as a parent message. BOSS accepts such messages from any internal process. If you want to enroll a job for execution under BOSS from a process running in parallel with BOSS, use the utility program 'bossjob'. 'bossjob' works like 'newjob' except that the message is sent explicitly to <:boss:> (instead of "the parent").

The utility program 'remotejob' even makes it possible to enroll a job for execution under a boss-process running in another RC8000 jobhost, but this presupposes that the job hosts are interconnected via a Disc to Disc File Router Interface (a DDFR-FDLC link), see ref. [13].

3.14 Mount tape

```
m+0:     14<12 + 0<5 + wait
 +2  ┐
 +4  ├    <:mount_:>
 +6  ┘
 +8  ┐
 +10 │
 +12 ├    <tapename>
 +14 ┘
```

The job requests a magnetic tape to be mounted. The message may be used both with and without wait indication:

wait = 0 means that the job does not want to wait,

wait = 1 means that the job wants to wait until the tape has been mounted.

ANSWER (BOSS): If wait = 1 and tapename = <::>, the job asks for a 'work tape' (see ref. [5] 2.130 (setposition, start positioning)). In this case, the name of the tape actually used is returned in the first 4 words of the answer. This name must be used in future references to the tape.

UTILITY PROGRAM: mount (ref. [7]).

IMPLICIT USE: This message may be sent by all data handling utility programs when input or output is on magnetic tape. Where magnetic tapes are handled in high-level languages, the message may be sent (with wait = 0) by 'open' and (with wait = 1) by 'setposition' or when a data transfer is checked. In all cases the rule is that the message is only sent if the tape is not already available.

PREREQUISITES (BOSS): The first mount-message concerning a certain tape costs a 'mount', which must be ordered when the job is started, see ref. [3] 3.1. Subsequent mount-messages concerning the same tape are free, as long as the tape has not been 'released' (see 3.22) or 'suspended' (see 3.20) by the job. Furthermore you have to order a 'station' or a 'special tape station' (by means of 'stat' or 'device', respectively) for each simultaneously mounted tape, see ref. [3] 3.1.

OPERATING SYSTEM ACTIONS:
BOSS: It is first checked, that the job has ordered a 'station' or a 'special station' for the tape. If the tape is not already mounted, and the operator has not already been asked to mount the tape, a request for the tape is issued.

If wait = 1, the job will be stopped and swapped out until the tape is mounted, possibly on a specific station (if this was previously ordered with 'mount special' (see 3.32)).

Next the job is included as a user of that device (peripheral process), where the tape is actually mounted, swapped in and started again.

The tape is now registered as being presently used by ('assigned to') the job, until the job either finishes, suspends the tape (3.20) or releases the tape (3.22). If the tape was not previously registered with connection to a special station (see 3.22) it will be registered as "standard".

ERROR REACTIONS (BOSS): The job will be aborted if:
   the tape is reserved for another project
   'stations' would be exceeded, or
   the tape is being used by another job.

FURTHER INFORMATION: Use of magnetic tapes in BOSS is described in ref. [3] 6.1, and in ref. [14] chapter 5. The handling of magnetic tape mounting in utility programs is described under the heading "kind = 18" in ref. [6] 6.2 and 6.5. The handling of magnetic tape mounting in high-level languages are described in ref. [4] 2.3.3.1 (Details of Handling of Device Status, Magnetic tape, Does not exist), ref. [5] 2.95 (Open, Initialization of a Document, Magnetic Tape and example 2) and in ref. [5] 2.130 (Setposition, Start Positioning and example 3).

3.16 Print

m+0 16<12 + pattern <5 + wait

This message provides for flexible communication with the operator for purposes which are not covered by the other parent messages.

Pattern may be anything and the message may be sent with or without wait-indication.

ANSWER (BOSS): If wait = 1, the first 4 words of the answer will contain the textword from the operator answer and the last word will contain the integer from the operator answer. If one (or both) is missing in the operator answer, it (they) will be replaced by NULL's.

UTILITY PROGRAM:

opmess sends the message with wait = 0.
opcomm sends the message with wait = 1.
Both are described in ref. [7].

IMPLICIT USE: The standard procedure 'open' (and all data handling utility programs) will send a 'print'-message with the text:

wait for <name of document>

and wait = 0, during the initialization of a reader or cardreader if the device is reserved by another job. Open then waits (busy waiting) until the device is free. See ref. [5] 2.95 (Open, Initialization of a document), and ref. [6] 6.2 (Connection of a file, kind 10 and 12).

OPERATING SYSTEM ACTIONS:
s: handles the message in the standard way. Notice: this implies that the operator can not supply an answer.
BOSS: displays the message to the operator. If wait = 1, the job will be stopped and swapped out until the operator answers the message displayed. The operator may add a textword (max. 11 chars) and/or an integer to the answer-command. (see ref. [4] chapter 3 ('answer'-command)).

These will be returned in the answer to the message, see ANSWER (above).

FURTHER INFORMATION: Notice, that there is a version of the standard procedure 'system', especially designed to send this message with pure text as contents (see ref. [5] 2.149 system(10, i,s)).

3.18 Mount ring

```
m+0        18<12 + 0<5 + 1
 +2 ┐
 +4 }      <:ring_:>
 +6 ┘
 +8 ┐
 +10│
 +12}      <tape name>
 +14┘
```

The job demands a write-enable ring on the tape.

ANSWER (BOSS): If the tape is already "known" to BOSS one way or another, the first 4 words of the answer contain the tapename and the last 4 words are undefined. Otherwise the answer is undefined.

UTILITY PROGRAM: Ring (ref. [7]).

IMPLICIT USE: This message may be sent by all data handling utility programs making output to magnetic tape. Where magnetic tapes are handled in high-level languages, the message may be sent when a data transfer (output) is checked, see ref. [4] 2.3.3.1 (Details of Handling of Device Status, Magnetic tape, Stopped).

PREREQUISITES (BOSS): If the message is to have any effect, the tape must previously have been requested by means of 'mount tape' or 'mount special' (with wait = 1, see 3.14 and 3.32), and not later suspended (see 3.20) or released (see 3.22).

OPERATING SYSTEM ACTIONS:

BOSS: The message may have no effect, as described above. The message also has no effect if the tape is actually already mounted and has a write-enable ring.

Otherwise, the job is first excluded as a user of the device (peripheral process), where the tape is mounted. Then the job process is stopped and swapped out, the tape is unloaded and the operator is asked to mount the tape again, with a write-enable ring, possibly on a specific station (if this was previously ordered by means of 'mount special (see 3.22)).

When the operator has mounted the tape again the job will be included as a user of that device where the tape is actually mounted, swapped in and started again.

Please notice that BOSS does not check whether the operator actually did mount the ring when he remounted the tape.

BOSS ignores the value of wait.

ERROR REACTIONS (BOSS): The job will be aborted if the tape is presently being used by ('assigned to', see 3.14) another job, or if the sending job is not permitted to use that particular tape with a ring, see ref. [3] 6.1.2 (access code).

FURTHER INFORMATION: See the other messages concerning magnetic tapes: 3.14, 3.20, 3.22 and 3.32.

3.20 Suspend tape 3.20

```
m+0        20<12 + 0<5 + 0
 +2  ┐
 +4  }     <:suspend_:>
 +6  ┘
 +8  ┐
 +10 │
 +12 }     <tape name>
 +14 ┘
```

The job releases the station on which the tape is mounted, but warns that the tape may be needed again later.

ANSWER (BOSS): As for 3.18 'mount ring'.

UTILITY PROGRAM: suspend (see ref. [7]).

IMPLICIT USE: Where magnetic tapes and flexible discs are handled in high-level languages, this message will be sent when the standard procedure 'close' is called like this: close(z,true), see ref. [5] 2.25 (close) and ref. [15] 3.6 (Releasing a Flexible Disc).

PREREQUISITES (BOSS): A suspend message concerning a 'work tape' (see ref. [5] 2.130 (setposition, start positioning)) costs a 'suspend', which must be ordred when the job is started, see ref. [3] 3.3.

OPERATING SYSTEM ACTIONS (BOSS): The message has no effect if the tape is not registered as being used by ('assigned to') the job.

Otherwise BOSS will do as follows (if relevant):
exclude job as user of the tape station,
std.station: becomes a "free" station,
std.station: decrease 'stations' used by one,
cancel operator requests concerning the tape,
work tape: increase 'suspends' used by one,
rewind the tape.

Notice, that the assignment of the tape to the job is not cancelled, neither is the station connection ('std' or 'special').

The tape may later be unloaded by BOSS or by the operator if the station is needed for another tape.

ERROR REACTIONS (BOSS): The job will be aborted if the tape is a work tape, and the job has no 'suspends' left.

FURTHER INFORMATION: Please notice that utility programs generally do not suspend or release tapes (see ref. [6] 6.3). The advantage is that a tape may be left correctly positioned for the next program, but then, if the tape is actually no longer needed, or the station is to be used for another tape, you have to call one of the programs 'suspend' or 'release' (ref. [7]).

3.22 Release tape

m+0	22<12 + 0<5 + 0
+2	<:release_:>
+4	
+6	
+8	<tape name>
+10	
+12	
+14	

The job releases the tape and the station on which it is mounted. If the tape is a work tape, it becomes public.

ANSWER (BOSS): As for 3.18 'mount ring'.

UTILITY PROGRAM: release (see ref. [7]).

IMPLICIT USE: As for 3.20 'suspend tape' except that the call of 'close' must be:

```
   close (z, false add 1).
```

PREREQUISITES: None.

OPERATING SYSTEM ACTIONS (BOSS): The message has no effect if the tape is not registered as being used by (assigned to) the job. Otherwise BOSS will do as follows (if relevant):

If the tape was not already suspended (see 3.20):
exclude job as user of the tape station,
std. station: becomes a "free" station,
std.station: decrease 'stations' used by one,
cancel operator requests concerning the tape.

Suspended work tape:
decrease 'suspends' used by one

In all cases:
cancel the assignment of the tape to the job (this makes a work tape public) and the station connection (std/special), rewind the tape.

The tape may later be unloaded by BOSS or by the operator if the station is needed for another tape.

ERROR REACTIONS: None.

FURTHER INFORMATION: As for 3.20, except that the utility programs 'save' and 'load', and the utility programs for flexible disc handling may send 'release' messages, see ref. [7] and ref. [15].

3.24 Load

```
m+0        24<12 + 0<5 + 0
 +2 ⎫
    ⎬      <:load_:>
 +4 ⎭
 +6
 +8 ⎫
+10 ⎪
    ⎬      <reader name>
+12 ⎪
+14 ⎭
```

The job requests the loading of a new tape in a paper tape reader or a new deck of cards in a card reader.

UTILITY PROGRAM: None.

IMPLICIT USE: The standard procedure 'open' (and all data handling utility programs) will send this message during the initialization of a paper tape reader or cardreader, and then wait (busy waiting) until the first non-empty data block is received from the device. See ref. [5] 2.95 (Open, Initialization of a Document) and ref. [6] 6.2 (connection of a file, kind 10 and 12).

PREREQUISITES (BOSS): The use of paper tape reader and card reader from a BOSS job is usually handled as 'job controlled reading', and permission to use this facility must be ordered when the job is started. A full description is given in ref. [3] 6.3 (Tape Reader) and 6.4 (Card Reader).

OPERATING SYSTEM ACTIONS:

s: notice that the normal implicit use has the effect that the job continues by itself as soon as the operator loads the reader in question.

BOSS: If <reader name> is not the name of a 'job controlled reader' (cf. PREREQUISITES), the message will be handled as described for 3.16.

Otherwise, the job will be stopped and swapped out while BOSS initializes the reading and instructs the operator. When the reader becomes ready the job will again be swapped in and allowed to continue.

3.26 Change paper

```
m+0        26<12 + 16<5 + 1
 +2 ⎫
 +4 ⎭      <:change:>
 +6        <paper type>
 +8  ⎫
 +10 ⎪
 +12 ⎬     <device name>
 +14 ⎭
```

The job demands new paper in a device, usually a printer or a punch.

ANSWER (BOSS): If <device name> = <:printer:>, the first word of the answer signals the result:

0 : ok, message accepted.

2 : message rejected, job controlled printing not ordered (cf. PREREQUISITES).

Otherwise the answer is as described for 3.16.

UTILITY PROGRAM: change (see ref. [7]).

IMPLICIT USE: This message may be sent by all data handling utility programs and by programs written in high-level languages when printer or punch is used, see ref. [6] 6.5 (Standard Recovery Actions, kind 12 and 14) and ref. [5] 2.3.3.1 (Details of Handling of Device Status, kind 12 and 14).

PREREQUISITES (BOSS): The use of a printer from a BOSS job is called job controlled printing, and permission to use this facility must be ordered when the job is started. A full description is given in ref. [3] 6.2 (Printer). The use of a punch must be ordered by means of 'device', see ref. [3] 3.1.

OPERATING SYSTEM ACTIONS (BOSS): If <device name> = <:printer, the job will be stopped and swapped out until the operator has changed the paper. Then the job is swapped in again and allowed to proceed.

Otherwise, the message is handled as described for 3.16.

ERROR REACTIONS (BOSS): See ANSWER.

3.28 Timer

```
m+0      28<12 + 3<5 + 0
 +2 }
 +4 }    <:timer_:>

 +12     <time1>: seconds to the interrupt
 +14     <time2>: seconds allowed the job to respond to the
         interrupt.
```

The job requests a 'parent break' when <time1> seconds have elapsed, and then to be given a respite of a further period of <time2> seconds to respond to the break.

The message may be used to make sure that a program does not use an unreasonable amount of computer time. It is especially useful during debugging, where a program may enter an endless loop.

A new 'timer' message cancels previously sent 'timer' messages.

UTILITY PROGRAM: timer (see ref. [7]).

OPERATING SYSTEM ACTIONS (BOSS): When <time1> seconds have elapsed, BOSS executes the 'parent break':

```
   stop job,
   store registers and instruction counter at the interrupt
         address of the job,
   set 'break cause' = 8 (parent break),
   restart the job in its 'interrupt handling routine'.
```

The job is now expected to respond by sending a 'break' message (see 3.4), which will cause BOSS to perform the 'dump and restart' action as described in 3.4. The job may also respond by sending a new 'timer' message, which will then cancel the present 'timer' message.

If, however, the job does not respond to the 'parent break' before <time2> seconds have elapsed, BOSS will perform the 'dump and restart' action described in 3.4 as if the 'break' message had been received - with one minor addition: when the process has been stopped, BOSS will store the present values of registers and instruction counter to replace the values stored by the 'parent break' (because there is obviously an error in the interrupt handling routine of the job, and the new values may help to find it).

ERROR REACTIONS (BOSS): The job will be aborted if its interrupt address is undefined.

FURTHER INFORMATION: See 3.4. BOSS: The elapsed time will be measured in 'real', ~~physical time as on your own grandfather clock, except that time spent~~ swapped out is not counted.

job consumed time

3.30 Convert

m+0	30<12 + 16<5 + 0
+2	<printerspec>
+4	
+6	<paper type>
+8	<file name>
+10	
+12	
+14	

The job reports that it wants to have a file printed on a certain line printer.

The interpretation of <printerspec> is the same as for 'newjob' (see 3.12), except that 0, 0 must be replaced with the text <:conv:>. This text is inserted by the utility program 'convert' (see ref. [7]), if it is called without a printer-parameter.

ANSWER (BOSS): The first word of the answer signals the result:
 0: ok, convert accepted
 1: cbufs exceeded
 2: file does not exist
 3: file has wrong scope
 4: temporary resources insufficient
 5: file in use
 6: file is not a backing storage area
 7: file is no text file (contents key is not 0)
19: attention status at remote terminal
20: device unknown
21: device not printer
22: parent device disconnected

UTILITY PROGRAM: convert (see ref. [7]).

PREREQUISITES (BOSS): Each convert costs a 'cbuf' which must be ordered when the job is started, see ref. [3] 3.3. As the default value is rather high (8), there is normally no need to bother about this.

The file must either be a permanent file (on any backing storage device) or a temporary file on the system disc (the backing storage named <:disc:>).

OPERATING SYSTEM ACTIONS (BOSS): If the message is accepted (see ANSWER), the request will queued up for execution when its turn comes. If the file is temporary it will be taken over by BOSS, i.e. the job will no longer be able to see the file, and the resources (the segments and the catalog entry) used to create the file will not be given back to the job.

ERROR REACTIONS (BOSS): The job will be aborted if it has no 'cbufs' left (cf. PREREQUISITES). See also ANSWER.

FURTHER INFORMATION: As BOSS prints directly from the file if it is permanent, you should avoid using, changing or removing it before your are positively certain that BOSS has finished the printing. Please notice the contrast between this, and the online command 'convert' which always makes a copy of the file and prints from the copy. Please also notice, that you can not use this message to convert a 'login' file.

There is a detailed example on how to send this message from an ALGOL program, in ref. [5] 2.149 (system, example 5).

3.32 Mount special

```
m+0       32<12 + 16<5 + wait
 +2  ⎫
     ⎬    <:mount_:>
 +4  ⎭
 +6       <device number>
 +8  ⎫
 +10 ⎪
     ⎬    <tape name>
 +12 ⎪
 +14 ⎭
```

The job wants a magnetic tape to be mounted on a specific tape station. The message may be used with or without wait-indication, but it is normally used without.

ANSWER: As for 'mount tape' (see 3.14).

UTILITY PROGRAM: 'mountspec' (sends the message with wait = 0 see ref. [7]).

IMPLICIT USE: The utility programs 'save' and 'load' will send this message as required, if they are called with the parameter 'mountspec'.

PREREQUISITES (BOSS): The tape station in question must have been ordered when the job was started. This is done by means of 'device', see ref. [3] 3.1. The tape must not be registered as being used by (assigned to) another job, or by the sending job on a standard station or a different device. The message may cost a 'mount' as described in the prerequisites for 3.14.

OPERATING SYSTEM ACTIONS (BOSS): As for 3.14 (mount tape), except that the tape will be registered with connection to the special tape station in question. This has the effect, that operator requests concerning the tape will automatically include the device number, even if you later on use 'mount tape' messages instead of 'mount special' messages. This connection will be cancelled by 'release' (see 3.22) but not by 'suspend' (see 3.20).

ERROR REACTIONS (BOSS): The job will be aborted if the prerequisites are not fulfilled:

not special station
station not ordered
tape used by other job
mount special after mount (the tape was already connected as 'std').

FURTHER INFORMATION: See the other messages concerning magnetic tape (3.18, 3.20, 3.22 and especially 3.14).

The usual use of this message is, that if you want to have a tape mounted on a special station, you only call the utility program 'mountspec' before the first mounting of the tape, and then again each time the tape has been 'released'. But notice that utility programs generally do not 'release' tapes, see 3.20.

3.36 Corelock

```
m+0      36<12 + 2<5 + 0
 +2 ┐
 +4 }    <:corelock_:>
 +6 ┘

 +12     <number of seconds in the period>
```

The job requests not to be interrupted, stopped or swapped out for a certain period of time.

UTILITY PROGRAM: corelock (see ref. [7]).

IMPLICIT USE: None.

PREREQUISITES (BOSS): The permission to use this facility, and the maximum duration of the period must be ordered at jobstart, and the permission must also be stated in the user catalog.

OPERATING SYSTEM ACTIONS (BOSS): If the request fulfils the prerequisites, the job will be locked in core, as if the job had infinitely high priority, for as many seconds as stated in the message. BOSS now expects to receive a 'coreopen' message (see 3.38) from the job before the stated period of time is over. When the message is received, the job resumes normal priority and proceeds.

If, however, no such message is received the job will be aborted.

ERROR REACTIONS (BOSS): The job will be aborted if:
  corelock is not allowed,
  corelock time is exceeded (coreopen not received).

FURTHER INFORMATION: See ref. [3] 6.7.3.

3.38 Coreopen 3.38

```
m+0        38<12 + 0<5 + 0
 +2 ┐
 +4 }      <:coreopen:>
 +6 ┘
```

This message is only used in connection with 'corelock', see the description in 3.36.

UTILITY PROGRAM: coreopen (see ref. [7]).

IMPLICIT USE: None.

PREREQUISITES: None.

ERROR REACTIONS: None.

OPERATING SYSTEM ACTIONS (BOSS): The message has no effect, except when used in connection with 'corelock', see 3.36.

3.44 Function 44 3.44

Not used at present. Will be treated as 3.16.

3.46 Function 46 3.46

Not used at present. Will be treated as 3.16.

3.48 Function 48 3.48

Not used at present. Will be treated as 3.16.

3.50 Create link

m+0	50<12 + 7<5 + 1
+2 +4 +6 +8	<devname> (in devicehost)
+10	<kind> < 12 + <subhostno> (at jobhost)
+12	<hostid> (devicehost)
+14	<homereg> < 12 + <netid>

The job requests the creation of a temporary (remote) link.

A special version of this message is used for TELEX devices (see ref. [2] 3.5)

<devname> is the name of the device (in the devicehost), which the link is to connect.

<subhostno> and <hostid> identify the endpoints of the link.

<homereg> and <netid> are not used at present, but both should be zero.

A. REFERENCES

[1] RCSL No 31-D628:
BOSS2 Installation and Maintenance Manual

[2] RCSL No 31-D474:
The TELEX facilities at MOT

[3] RCSL No 42-i1265:
BOSS2 User's Manual

[4] RCLS No 42-i0781:
ALGOL7 User's Manual, Part 1

[5] RCSL No 42-i1278:
ALGOL8 User's Guide, Part 2

[6] RCSL No 31-D364:
SYSTEM 3 UTILITY PROGRAMS, Part One

[7] RCSL No 31-D590:
SYSTEM 3 UTILITY PROGRAMS, Part Two

[8] RCSL No 31-D379:
SYSTEM 3 UTILITY PROGRAMS, Part Three

[9] RCSL No 31-D552:
RC8000 OPERATING SYSTEMS: INTRODUCTION

[10] RCSL No 31-D643:
OPERATING SYSTEM s, Reference Manual

[11] RCSL No 31-D476:
RC8000 MONITOR, Part 1, System Design

[12] RCSL No 31-D477:
RC8000 MONITOR, Part 2, Reference Manual

[13] RCSL No 31-D625:
Remotejob Utility Program

[14] RCSL No 31-D498:
BOSS2 Operator's Manual

[15] RCSL No 31-D459:
Flexible Disc Handling within the ALGOL System,
User's Manual

[16] RCSL No 31-D596:
Utility Programs for Flexible Disc Handling

[17] RCSL No 31-D557:
CREATELINK, RELEASELINK, LOOKUPLINK, LOOKUPDEV
Utility programs

RETURN LETTER

Title: Parent Messages in RC8000 RCSL No: 31-D610

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

____

____

____

____

Do you find errors in this manual? If so, specify by page.

____

____

____

____

How can this manual be improved?

____

____

____

____

Other comments?

____

____

____

____

____

Name: ____________________ Title: ____________________

Company: ______________________________________

Address: ______________________________________

Date: __________

Thank you

. . . . Fold here . . . .

. . . . Do not tear - Fold here and staple . . . .

Affix
postage
here

RC REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark

information	repl. JMA 820813		ident TSH 830216		page 1/5
	RC4000	RC8000			class EXT

subj.
Corrections to RCSL No 31-D610: Parent Messages in RC8000

This RC Information Note contains corrections to "Parent Messages in RC8000" (RCSL No. 31-D610, October 1981).

1. (NEW) Page 2:

Replace the line "m+0: ...", with
m+0: function < 12 + pattern < 5 + aux < 1 + wait

Before the section "wait may be zero ...", insert:

aux is 4 bits containing additional information concerning the interpretation of the message. At present the value is irrelevant for all other messages than "newjob" (see 3.12).

2. (NEW) Page 3: (only valid for monitor release 9.0)

Bottom, replace last line with:
except 'finis', 'replace' and 'newjob', see 3.2, 3.10 and 3.12.

3. Page 4:

Midpage, after "...should be used anyway" add: (with the exception mentioned in 2.4).

Bottom, insert: Notice that in this case you have to specify WAIT = 1, if you want to be able to check the answer; this is because 'system' only returns the answer in your array if WAIT = 1.

4. (NEW) Page 5: (only valid for monitor release 9.0)

OPERATING SYSTEM ACTIONS, replace the action of s with:
s: the job process will be removed. See ref. [10] 2.2.

5. (NEW) Page 10: (only valid for monitor release 9.0)

PREREQUISITES, add
s: The replacement file must be visible from the sending process.

6. (NEW) Page 11: (only valid for monitor release 9.0)

OPERATING SYSTEM ACTIONS, cancel "if wait = 1", in the action og s.

7. (NEW) Page 12:

Replace the line "m+0 ...", with:
m+0 12<12 + 0<5 + aux<1 + 1

Replace the section "PREREQUISITES ...", with:
PREREQUISITES (s): The job file of the new job must be visible from the catalog base of the sending process (only valid for monitor release 9.0).

PREREQUISITES (BOSS): The job file of the new job must be permanent, and it must be visible from the standard base (aux=0) or the catalog base (aux=1) of the sending job.

8. (NEW) Page 13: (only valid for monitor release 9.0)

OPERATING SYSTEM ACTIONS, add
s: When s recaives the message, an answer with result = 1 is returned and s starts the execution of the commands specified in the job file. Messages from the exection (~~execution~~ error messages or output from 'list' or 'max' commands) will be printed on the console of the sending process. <printerspec> is ignored.

9. Page 31:

Replace ERROR REACTIONS .. with:

ERROR REACTIONS (BOSS): See ANSWER.

10. Page 41:

Replace the first page of the description with:

```
m + 0  50<12 + 7<5 + 1
  + 2 ⎫
  + 4 ⎬ <device descriptor>
  + 6 ⎪
  + 8 ⎭
  + 10 <kind> <12 + <subhostno> (at jobhost)
  + 12 <hostident> (devicehost)
  + 14 <homereg> <12 + <netid>
```

The job requests the creation of a temporary (remote) link.

<device descriptor>: The interpretation depends on <kind>:
   <kind> = 0: <device descriptor> is supposed to be the name of a catalog entry which is then used to create the link as described in ref. [3] 7.2.

   <kind> = 0 and first word of <device descriptor> = 1 works as <kind> = 14 and <device descriptor> = <:printer:> (see below).

Otherwise, <device descriptor> is used directly, and this should normally be the device host's name for the device. If kind = 36 (telex) <device descriptor> has a special format as described in ref. [2] 3.5.

<subhostno> and <hostident> identify the endpoints of the link. If they are not supplied (both = 0), the values from your terminals link will be used (this presupposes that your terminal has a temporary link).

<homereg> and <netid> are not used at present, but both should be 0.

11. Page 42:

Function results, replace result 0 with:

0: ok, link created as requested. In this case the remaining part of the answer contains:

```
word 2-5: the name of the link process created
word 6  : name table address of the link process.
word 7  : mode <12 + kind
word 8  : link buffer size
```

replace result 7 with:
7: (see below).

Replace result 9 and 10 with:

7: device has permanent link

Device already has a temp (remote) link which is:
20: reserved by BOSS
21: reserved by some other process

30: links exceeded (job not allowed to use links)

Link creation ok, but job could not be included as a user of the link process; function result equals 30 + result of 'include user':

32: BOSS not user

33: job (!) does not exist (should not occur)

34: device number does not exist (should not occur)

BOSS received a dummy answer to the message to <:host:>; function result equals 40 + 'answer result':

42: rejected

43: unintelligible

44: malfunction

45: does not exist

After creation of the link, BOSS temporarily reserves the link process (it will be released again before the job is answered). If this reservation fails, function result equals 50 + result of 'reserve process':

(51: reserved by other (see 21))

52: BOSS not user

53: Link process does not exist

12. Page 43:

Replace ERROR REACTIONS .. with:

ERROR REACTIONS (BOSS): See ANSWER.