P/N 99103250

RCSL No: 30-M329 PN 99103250

Edition: January, 1983

Author: Henning Jakobsen

Title:

RC3502 - TOP35 - Test Operating System
User's Guide

RCSL No 42-I 2164

RC International

Keywords:

RC3502, Test Operating System, TOP35, REAL TIME PASCAL, English Language, RC8000.

Abstract:

This manual describes how to use TOP35, the test operating system for the RC3502 computer. After loading TOP35 on an RC3502 you can manage (initialize, start and stop) the different testprograms, which are used to test the RC3502 hardware (CPU, Memory, I/O-controllers, etc.).

(36 printed pages).

Printed by A/S Regnecentralen af 1979, Copenhagen

RCSL No 42-I 2164

TABLE OF CONTENTS PAGE

1. INTRODUCTION 1.

This manual will guide you through your work with the RC3502 testprograms. The manual describes how to load the RC3502 and what to do afterwards. This includes a complete command syntax and examples showing how the different testprograms are handled.

A more detailed description about handling the individual testprograms is to be found in the respective manuals (see appendix A).

TOP35 is a Test OPerating system developed for the RC3502 computer.

TOP35 is programmed in Real Time PASCAL. The system design is based on the ideas behind TOP, which was developed for the RC8000 computer.

2. THE TOP35 SYSTEM PACKAGE FOR RC3502 2.

The TOP35 system consists of a main process "top" and an underlaying testprocess for each running test. An example is shown in fig. 1.

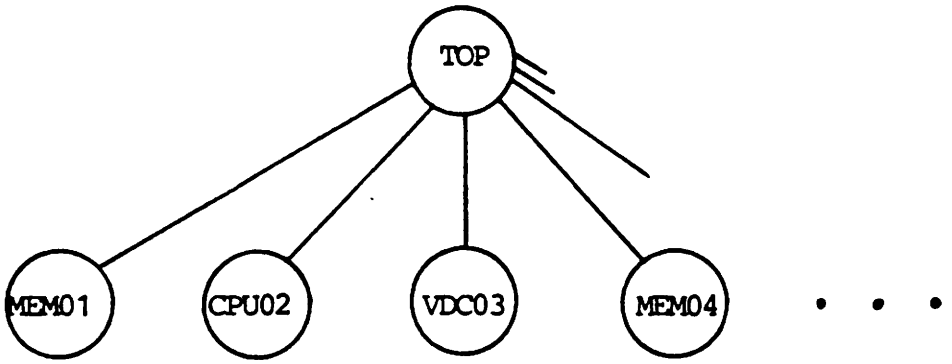

Figure 1: Snapshot of process hierarchy.

The real system is a bit more complicated. That is, each test consists of three or more processes: The test itself, and a test--program-server (TPS). Moreover, the test could have privat drivers and other underlaying processes. Your attention is drawn to these facts only because lack of memory and system faults could produce errors (exceptions) which can only be traced to these abbreviated names.

The TOP35 system is delivered both as files on a magnetic tape (for installations which include an RC8000 with a magnetic tape station) and programmed in EPROMs (for installations where TES201/202-modules can be used).

For a precise description of the contents of the package, you have to read appendix C or newest SW2201 Package Description.

3. LOADING TOP35 INTO RC3502 3.

The TOP35 system may be loaded into RC3502 as a normal process hierarchy under 'opsys'. This can be done by autoloading from RC8000 via the FPA-link to RC3502. If you hold the system in EPROMs (i.e. a TES201/202-module), you can also autoload without an external load-device (RC8000).

3.1 Preparing Autoload from RC8000 3.1

If you want to autoload the RC3502 with a so-called bootfile via an FPA-link (from RC8000) take the following action on the RC8000.

First you have to put your bootfile(s) on the disc. This can be done by loading the delivered magnetic tape with the system utility program LOAD (RCSL No 31-D491). If you look at appendix C or the newest SW2201 Package Description, you can see the available bootfiles and jobfiles. The jobfile is used to autoload the RC3502 with the associated bootfile.

Please note that more files are present in later versions.

You may check the existence of the files by performing the following commands:

```
att s
new juul run
lookup rc3502top rc3502mem rc3502cpu rc3502vdc rc3502mir
lookup autotop automem autocpu autocom autohlc autovdc
       automir
```

You can load the magnetic tape as a BOSS-job or in an "s"-process.

Example:

```
att s
new juul base -8388607 8388605 run
t = set mto mtsw2201 0 2
i t
```

In this example the files from the third file of the magnetic tape is loaded with system scope.

Second you have to find the name of the main process running the FPA to the RC3502 in question. This name is main35021 in the jobfiles. If your name is different (ask the operator), you have to change it.

At last you have to start the autoload. This is done by the utility program AUTOLOAD (RCSL No 31-D471). You can either use the BOSS-job on the jobfile associated with your bootfile or the following in an "s"-process:

```
att s
new juul run
main35021=autoload rc3502top start no.
```

Before starting the job you have to check the switches on the CPU (see fig. 2 and the next section) and press the autoload button on RC3502, Power Panel.

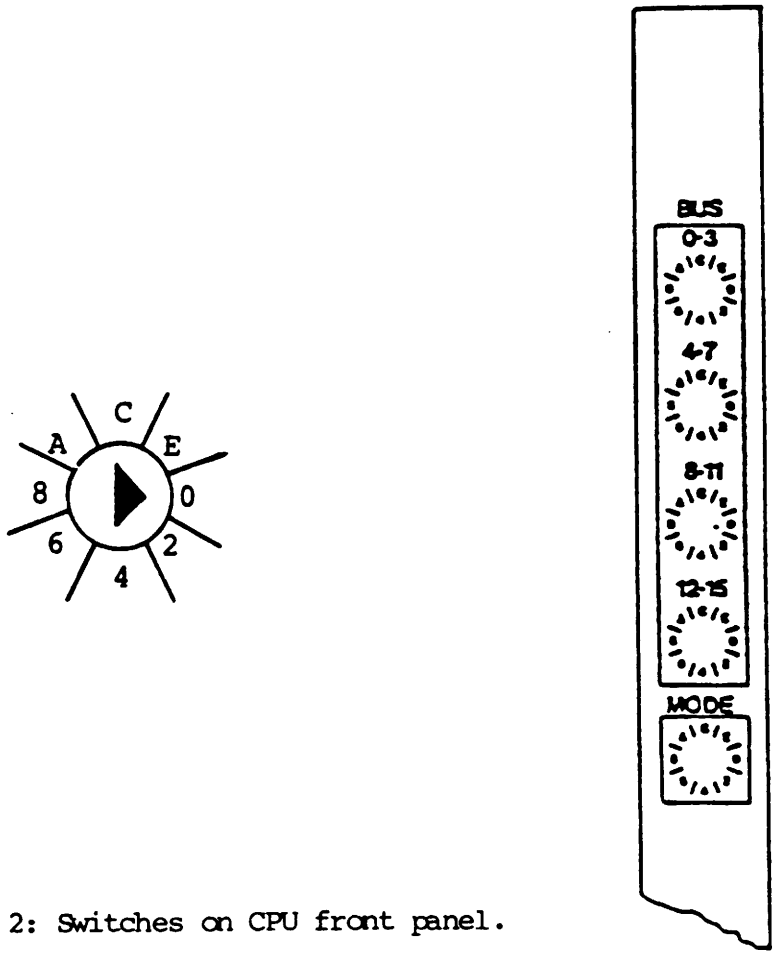

Figure 2: Switches on CPU front panel.

3.2 Autoload of RC3502

3.2

If you have an autoloadjob ready to be started on RC8000 or if the testsystem is found in one or more TES201- or TES202-modules, do the following:

1. Turn on power on RC3502.
2. Set the first switch (Bus 0-3) on front of the CPU panel (see fig. 2) to detect your load device:
   FPA100 (RC8000): A
   TES201/202 (RC3502): E
3. Press autoload button and start the autoloadjob if autoloading from RC8000.

If your autoload medium is a TES201/202, you have to insert them before turning on the power. Remember to set the switches on the TES201/202 to an unused module number.

3.3 Output after Autoload 3.3

Autoload from FPA in channel 86 (decimal), switches = A056.

When you press the autoload button, the following picture will be shown on the debug console:

```
RC3502 MICRO VER05
* DEBUG VER02
*R
T

Autoload from fpa in 0056H
```

Figure 3: Autoload start picture.

After a succeeded autoload, the following will be shown too:

```
RC3502 MICRO VER05
* DEBUG VER02
*R
T

Autoload from fpa in 0056H
.....................................................................

>opsys
rc3502 pascal80 release : 5.03 1982.12.09
>top
RC3502-TOP80 Test Operating System. Rev. 2.00 1983.01.27
=============================================================
RAM at module no. :    c0   c2   c4
ROM at module no. :    e0
Select function:
```

Figure 4: Picture after succeeding autoload of TOP35.

4. HOW TO USE TOP35 4.

After loading the TOP35 system into the RC3502, you are able to communicate with three different types of processes via the debug console. These are shown in sections 4.2, 4.3 and 4.4. In section 4.1 you can see how to select the process you want to communicate with. In the shown examples, underlining always means your input.

4.1 Communication via Debug Console 4.1

At first the console must be in terminal mode (T-mode). There are two possible modes: T-mode and D-mode (debug mode). Mode switching is done by activating the BELL (i.e ctrl G) key (fig. 5). After autoload the console starts in T-mode, as seen from fig. 4 in the previous chapter.

```
D
* ctrl G
T
```

Figure 5: Mode switching.

All input and output are identified with a name (e.g. displayed as: >top). The console will always remember the last output or input name.

If you want to communicate to the process, which name is the last printed on the console, you type your text followed by a RETURN (see fig. 6a).

You can type only one line at a time. Each line is ended by a RETURN.

It is always possible to write the name of the process after activation of the ATTENTION key (i.e. esc). The name must be ended by a RETURN. The following line will be accepted as input to this process (see fig. 6).

For a complete description of this mechanism, see ref. [1].

```
>top
RC3502-TOP80 Test Operating System. Rev. 2.00 1983.01.27
==========================================================
RAM at module no. :    c0   c2   c4   c6
ROM at module no. :    e0   e2
Select function:
display CR          <------------------------------------ a
Display      no test initiated.
Select function:
new:vdc CR          <------------------------------------ a
vdctest      initiated as vdc01
Select function:
>vdc01
-- vdc 201 test ---- ver 83.01.31 --
testprogram :       a
a: mirror in rc3502
>vdcprint01
select info:
>vdc01
b: terminal mirror
Select test:
ESC >top CR         <------------------------------------ b
Select function:
disp CR
Display
TEST  SUB-  RUN   ERRORS     DRIVER-INFORMATION     TEST
      TEST  NO.  DETECTED ACCESSED UNANSWERED LAST STATUS
==========================================================
vdc01          0      0         0          0      0 STOPPED ( 1)
Select function:
ESC >vdc01 CR       <------------------------------------ b
Select function:
list CR
-- vdc 201 test ---- ver 83.01.31 -- LIST OF PARAMETERS :
p  0  testprogram        :      a
p  1  no of runs         :     20
p  2  module no (master) :     16
p  3  channel            :     -1
p  4  module no (mirror) :     18
p  6  datacheck          :    yes
p 10  min blocksize      :      1
p 11  max blocksize      :    254
p 18  data kind          :      4
p 49  max message        :     10
Select function:
start CR
Select function:
```

Figure 6: How to change name of a process, to which input is typed in. CR is the RETURN key.

4.2 Creating the "top" Process 4.2

In a normal situation, "top" is automatically created and started as a child of a process adam. Process adam is started by the system at autoload.

This section is relevant only when the whole TOP35 system has failed. Then you can use the following commands instead of autoloading again. The process which serves the commands is "opsys". Therefore you have to start with esc opsys.

Input to opsys can be:

```
excode 47
break top
remove top
run s
```

The first 3 lines given above are also recommended when you want to get rid of top and all its childs.

Further notices on this can be found in ref. [1].

```
>opsys
excode 47
break top
>exception
1982.11.29 09.29
contest     >> exception , excode = 002f: break by father
gf = 00c4.2900 , top = 00c4.34f8 , code = 0010
called from: con01       , ic = 00c2.9fa4, line 1267 -1281 1983.01.28 10.23
>opsys
remove top
run s
>top
RC3502-TOP80 Test Operating System. Rev. 2.00 1983.01.27
==================================================
RAM at module no. :    c0   c2   c4
ROM at module no. :    e0
Select function:
```

Figure 7: Example showing a fault and how to repair.

4.3 Commands to "top" 4.3

When "top" is created and started, it will print an identification, which includes the date and versionnumber (see fig. 8).

```
>top
RC3502-TOP35 Test Operating System. Rev. 2.01 1983.07.27
Select function:
```

Figure 8: TOP35, headline information.

Every time "top" is ready to accept commands, the line "Select function:" will be printed. It is possible (but not recommended to beginners) to input more than one function in one line. Space is a legal separator of functions. If a function needs input (e.g. an answer or a value), you have to place that input before the next function in line, otherwise the next function will be interpreted as that input or just skipped.

The following functions are available, both in upper- and lowercase:

NEW:
The function has two variants:

a) Syntax: new:<testname>
   Action: A new test of the type <testname> will be initiated if possible. The test will ask for more information if necessary before start.
   Example:

```
Select function:
new:mem
mem initiated as mem01
Select function:
```

b) Syntax: new RETURN
   Action: All possible tests are printed.
   After the last line "Select test": you have to type a <testname>. The action is the same as in a).

Example:

```
Select function:
new
The following tests are available  :
vdctest
comtest
memtest
cputest
hlctest
imstest
Type the first 3 letters to select test.
Select test:
vdc
vdc initiated as vdc02
Select function:
```

DISPLAY:

Syntax: display

Action: A list of all initiated tests and their state will be printed.

Example:

```
Select function:
disp
Display
TEST  SUB-  RUN   ERRORS     DRIVER-INFORMATION     TEST
      TEST  NO.  DETECTED ACCESSED UNANSWERED LAST STATUS
=============================================================
mem01  c       7      0       196          0      0 RUNNING ( 6)
hlc02          0      0         0          0      0 STOPPED ( 1)
Select function:
```

BREAK:

Syntax: break:<versionname>

Action: If the test which was given the <versionname> when initiated, is running then this BREAK acts like a BREAK typed directly to that test.
This function is necessary when you cannot use the BREAK directly to the test (i.e. if the WAIT has been used).

Example:

```
Select function:
break:mem02
Break send to mem02
Select function:
```

KILL:

Syntax: kill:<versionname>

Action: The test which was given the <versionname> when initiated is broken and removed. Note that this can cause an error, if the test has started a driver, which at this moment holds an I/O-channel. The I/O-channel is never released. Most of the test will exit through an exception "break_by father".

Example:

```
Select function:
kill:com02
>exception
1982.11.29 09.34
contest     >> exception , excode = 002f: break by father
gf = 00c4.4c18 , top = 00c4.5810 , code = 0010
called from: com02       , ic = 00c2.9fa4, line 1267 -1281 1983.01.28 10.23
>top
com02        killed by TOP.
Select function:
```

HELP:

Syntax: help

Action: A list of functions and their parameters is printed.

Example:

```
>
help
Help :
The following functions are available in TOP :
  New         write : NEW:<testname>      Initiate a test.
  Help        write : HELP                Produce this list.
  Break       write : BREAK:<versionname> Break this test.
  Kill        write : KILL:<versionname>  Remove this test.
  Display     write : DISPLAY             Display test-status.
  Errorstat.  write : ERRORS              Display statistics.
<testname> is :vdc  com  mem  cpu  yyy  xxx
<versionname> is <testname> followed by 2 digits
Note that commands may be written with small as well as capital letters.
Input and output to/from a specific test is identified by <versionname>
Select function:
```

ERRORS:

Syntax: errors

Action: For each test is printed a line with name, run counter, error counter, and four lines with the 32 statistical counters.

Example:

```
errors
Errors:
mem01          runs          2     errors      0
        0      0       0       0       0       0       0       0
        0      0       0       0       0       0       0       0
        0      0       0       0       0       0       0       0
        0      0       0       0       0       0       0       0
hlc02          runs          1     errors      0
        0      0       0       0       0       0       0       0
        0      0       0       0       0       0       0       0
        0      0       0       0       0       0       0       0
        0      0       0       0       0       0     575     575
Select function:
```

4.4 Commands to the Test

When a new test is initiated by "top" (the NEW-function), information about the test, the versiondate and the names of the subtests is printed.

```
>mem01
-- mem 204 test ---- ver 82.12.30 --
testprogram        :a
a: address test
b: bit selection
c: complement test
d: jump test
e: long reliability
f: eprom checksum
Select test:
```

Figure 9: Test headline information.

Every time the test prints the line "Select test:" or "Select function:" the test is ready to accept input, but sometimes some functions are not valid.

As to "top" you can input a string of functions in one line, as long as each function is followed immediately with its parameters, if any.

Note that in many situations you will get the request "Select function:" before your request has been processed. That is because some requests will take an unknown timeperiod to process and other requests can be processed in the meantime.

The following functions are available both in lower- and upper-case as answers to the request "Select function:":

LIST:

Syntax: list

Action: All the parameters of the test will be listed with number, name and last value assigned (actual value).

Restricted use: No restriction.

Example:

```
Select function:
list
-- vdc 201 test ---- ver 83.01.31 -- LIST OF PARAMETERS :
p  0  testprogram          :      a
p  1  no of runs           :     20
p  2  module no (master)   :     16
p  3  channel              :     -1
p  4  module no (mirror)   :     18
p  6  datacheck            :    yes
p 10  min blocksize        :      1
p 11  max blocksize        :    254
p 18  data kind            :      4
p 49  max message          :     10
Select function:
```

PARAM:

Syntax: param

Action: The parameters will be listed as in the LIST-function, but after each actual value, you may assign a new value, before the next parameter is listed.
If you want to keep the old value, answer with a RETURN.

The new value may be a decimal number or a hexadecimal number. A hexadecimal number is typed as a 'h' followed by 1-4 hex digits (0..9, a..f).

Restricted use: Can be used only before START and after termination (i.e. not while the test is running).

Example:

```
Select function:
param
p  0  testprogram          :      a:
p  1  no of runs           :     20:100
p  2  module no (master)   :     16:
p  3  channel              :     -1:
p  4  module no (mirror)   :     18:h50
p  6  datacheck            :    yes:no
p 10  min blocksize        :      1:
p 11  max blocksize        :    254:25
p 18  data kind            :      4:
p 49  max message          :     10:3
Select function:
```

p<number>:

The function has two variants:

a):

Syntax: p<number>:<new value>

<number> is the number of the parameter to be changed.

<new value> is the value to be assigned to that parameter.

Restricted use: Not while the test is running (as in PARAM).

Action: The parameter gets the new value.

Example:

```
Select function:
p049:10
Select function:
```

b):

Syntax: p<number> RETURN

Action: The parameter is listed and updated as in the PARAM-function.

Restricted use: Not while the test is running (as in PARAM).

Example:

```
Select function:
p049
p049  max message          :     10:2
Select function:
```

START:

Syntax: start

Action: The parameters (which may have new values assigned) are checked and the test is started. If the test is already running, it is first BREAK'ed and then restarted.

Restricted use: No restrictions.

Example 1:

```
Select function:
start
Select function:
run no.     1              1983.04.05 09.27.20
```

Example 2:

```
run no.     3
run no.     4
run no.     5
start
run no.     6
--- Break and restarted.
run no.     1
```

BREAK:

Syntax: break

Action:
a) If the test is running, it will be terminated.
b) If the test is not started or already terminated, the parameters of the test will be assigned to their default values.
c) If b) is done, the next BREAK will force the test to its initialization fase, where the subtest can be selected.

Restricted use: No restrictions.

Example:

```
run no.     27
break                              <-------------- a
-- BREAK

--  Test terminated.
-- com 204 test ---- ver 83.03.18 -- LIST OF ERRORS :
--------------- run no.    60  : ---------------
No errors detected by testprogram.
----------------------- 1983.03.24  16.36.15 ----
Select function:
break                               <-------------- b
-- BREAK
   Test-parameters set to default.
Select function:
break                                <-------------- c
-- BREAK
-- com 204 test ---- ver 83.03.18 --
testprogram:        a
a: normal test
b: mirror only
c: testloops
Select test:
```

ERRORS:

Syntax: errors

Action: The actual run number of the test and information about detected errors is printed.
If the test has been terminated, the displayed information is as it was at termination time.

Restricted use: No restrictions, but if used before the first START, the information is of no use.

Example:

```
Select function:
errors
-- vdc 201 test ---- ver 83.01.31 -- LIST OF ERRORS :
--------------- run no.     2  : ---------------
     3 of type data field too long for buffer
    26 of type error in event code generation
Total number of errors :    50
----------------------- 1983.04.05 09.33.50 ----
Select function:
```

CHANGE:

Syntax: change

Action: The available kinds of alternative output devices are shown, and one is to be selected by typing the device name after the request "Select output device:".

Restricted use: No restrictions.

Example:

```
Select function:
change
No alternative outputdevice implemented yet.
Select function:
```

HELP:

Syntax: help

Action: A list of functions and their parameters is displayed.

Restricted use: No restrictions.

Example:

```
Select function:
help
Help :
The following functions are available in the edit-fase :
  Start       write : START           The test becomes running.
  Param       write : PARAM           List and update all parm.
  P<no>       write : P<no>=<value>   Update this parameter.
  List        write : LIST            List all parameters.
  Wait        write : WAIT            Read rest of line later.
  Help        write : HELP            Produce this listing.
  Break       write : BREAK           Terminates the test now.
  Change out  write : CHANGE          --- NOT IMPLEMENTED YET.
  Errorstat.  write : ERRORS          Output errorstatistics.
<no> is the number of the parameter you wish to change.
<value> is the new value of the parameter.
Note that commands may be written with small as well as capital letters.
Input and output to/from a specific test is identified by <versionname>
Select function:
```

WAIT:

Syntax: wait

Action: Stop the reading of input as long as the test is running. To be used between two STARTs, so the second will not be read before the test is terminated.

Note that after reading the WAIT no input (not even a BREAK or another WAIT) will be processed.

This feature is useful for setting up a line of commands and go home, while the test is running all night.

Restricted use: Only useful while the test is running.

Example:

```
Select function:
p2=hc4  start  wait  p2=hc6  start

-- test area locations  ( hex )  ee60  ..  ffff
-- test area size-1                2255
run no.      1           1983.03.24  16.24.40
run no.      2
run no.      3            <---------- input impossible now
run no.      4
--  Test terminated.
-- mem 204 test ---- ver 83.03.24 -- LIST OF ERRORS :
---------------- run no.     4  : ----------------
No errors detected by testprogram.
------------------------ 1983.03.24  16.24.51 ----
Select function:

-- test area locations  ( hex )  0000  ..  ffff
-- test area size-1                32767
Select function:
run no.      1           1983.03.24  16.25.01
run no.      2
run no.      3
run no.      4
--  Test terminated.
-- mem 204 test ---- ver 83.03.24 -- LIST OF ERRORS :
---------------- run no.     4  : ----------------
No errors detected by testprogram.
------------------------ 1983.03.24  16.27.52 ----
Select function:
```

MONITOR

Syntax: monitor: <number>

Action: Updates the variable monitor.
This variable controls extra output showing the u-fields of messages.

1	gives u-fields of errornous driver answers
2	gives u-fields of all driver answers
4	gives u-fields of all driver requests
256	gives u-fields of test requests.

These values may be combined.

5. ERRORMESSAGES FROM TOP80

The messages can be divided into 4 groups, depending of their origin.

5.1 System Messages

These messages come from the RC3502 system or they are propagated by general routines in the TOP80 system. This kind of errors has a catastrophic influence on the part of the TOP80 system from which they are propagated.

Exceptions:

If a program tries to do an illegal instruction (divide by 0, starting a new process without enough core, ...) the system calls an exception. This gives you a long trace of output, starting with an explanation of the cause.

```
vdctest      >> exception , excode = 001f: no core
gf = 0002.2000 , top = 0002.2350 , code = 00f4

called from: _initpool_rc, ic = 0001.a2fb, line 0-53   1981.02.03 11.53
called from: vdc02       , ic = 0001.022e, line 192-194   1981.03.17 19.07
```

Figure 10: Example of an exception caused by an error "no core" in process vdc02.

The exeption "no core" means that you have to add memory to the RC3502 or remove some other processes to get more free core.

Other exceptions could occur. The exceptions are described in ref. [1].

Unknown Name:

The mechanism which reads input names (i.e. ATTENTION followed by a name), will produce a message, if no process is waiting for input with that name.

```
>top
unknown name
```

Figure 11: Example showing the reaction on an unknown name.

This message will also occur if you are working too fast at the keyboard. Wait a moment, before retrying.
All possible names may be shown by
>opsys
list

System is Incomplete:

When a process in the TOP35 system fails in creating or removing a link to the code of another process or in creating an incarnation of another process, this message appears. The meaning if the linking fails, is that the process to be linked is missing. If the creation fails with result=3, the cause is lack of memory (as exception "no core").

```
>hlc01
Select function:
start
Select function:
?? SYSTEM IS INCOMPLETE :     Linking of  hdlc2        result :     1
?? SYSTEM IS INCOMPLETE :     Linking of  hdlc01driver result :     2
Select function:
```

Figure 12: Example of a failure, where a driver is missing.

5.2 Messages from "top" 5.2

Errormessages from "top" are displayed with a prefix.
Either: ? Operator error:
or !!! TOP35 error:

All errors of the second type indicate a system error or a strong misuse of the system. These kinds of errors should be reported.

The following messages can be expected:

? Operator error: unknown name :abc

a) <testname> specified after NEW: is unknown or missing. Write only NEW to get all legal <testname> displayed.

b) The <versionname> after KILL: is unknown or missing. Note that <versionname> consists of a three-letter long <testname> followed by a two-digit number, given to the test at initialization. Use DISPLAY to find the correct <versionname>.

? Operator error: Only 6 tests must be initiated concurrently.
This new:mem is rejected.

You have initiated too many tests. KILL some of them and try again.

? Operator error: this input not allowed now, no room.

The testprogram catalog in top is full. Break, remove, and run top again. Use only relevant NEW commands.

The following tests are available:
vdctest
comtest
memtest
cputest
hlctest
imstest
Type the first 3 letters to select test.
Select test:

This output is shown, if you write only NEW. The next input should be one of the shown <testname>.

mem initiated as mem01.

A memory test (mem) is initiated and given the <versionname> mem01.

Break send to mem01

If the test with the name mem01 was running, it is breaked.

mem01 killed by TOP.

The test running under the name mem01 is removed immediately.

5.3 Messages from a Test 5.3

The tests are able to display different types of messages:

a) errors done by the operator (prefixed with ?Operator error:)
b) system errors (prefixed with: !!!TOP35 error:)
c) ordinary messages (prefixed with -)
d) errors detected by the running test (prefixed with *)

The messages of type d) and some of type c) are described in the manual of the actual test. Here all general messages are shown. Errors of type b) should not occur while operating the TOP80 system in an ordinary manner.

? Operator error: this input not allowed now.

You are not allowed to use:
- the wait-function, if the test is not running (i.e. START'ed).
- the "update"-function (P<no> or P<no>:<value> or PARAM) if the test is running.

? Operator error: input not allowed under break-sequence.

Do not send more input to a test, which is trying to BREAK. This includes the BREAK, which can be sent from "top", if you use the

BREAK-function in "top". Input is allowed again, when the test is terminated.

? Operator error8 unknown parameter: p027

You are not allowed to change the value of this parameter.

? Operator error: illegal value.

You have tried to assign an illegal value to a parameter. Try again. Note that the subtest selection can give this output, if you select an unknown subtest.

6. AN EXAMPLE ON THE USE OF TOP35

6.

```
RC3502 MICRO VER05
• DEBUG VER02
•R
T

Autoload from fpa in 0056H
.....

>opsys
rc3502 pascal80 release : 5.03 1982.12.09
>top
RC3502-TOP80 Test Operating System. Rev. 2.00 1983.01.27
==========================================================
RAM at module no. :    c0   c2   c4   c6
ROM at module no. :    e0
Select function:
MEU:hlc
hlctest      initiated as hlc01
Select function:
>hlc01
-- com 204 test ---- ver 83.01.31 --
testprogram:        a
a: normal test
b: mirror only
Select test:
a
>hlcprint01
select information :
>hlc01
Select function:
param
p  0  testprogram          :      a:
p  1  no of runs           :     20:10
p  2  level no             :     72:
p  4  channel ( 0,1,both):       2:
p  6  datacheck            :    yes:
p  9  statuscheck          :    yes:
p 10  min blocksize        :      1:120
p 11  max blocksize        :    254:180
p 12  framegap (10 u_sec):       0:6
p 13  check modem state    :     no:
p 17  measure linespeed    :     no:yes
p 18  data kind            :      4:
p 20  timeout (100 msec)   :     30:
p 21  retry count          :      5:
p 49  max message          :     10:3
Select function:
start
Select function:
--- maximum test buffer size    :          180
--- maximum queue depth for xfer:            2
run no.      1
the measurred linespeed of channel        0  is  :     64kbps
the measurred linespeed of channel        1  is  :     64kbps
channel  :    0  connected

channel  :    1  connected

run no.      2
run no.      3
run no.      4
run no.      5
>top
disp
Display
TEST  SUB-  RUN   ERRORS     DRIVER-INFORMATION     TEST
      TEST  NO.  DETECTED ACCESSED UNANSWERED LAST STATUS
----------------------------------------------------------
hlc01  a      5      0      2199         7      1 RUNNING ( 6)
Select function:
>hlc01
run no.      6
run no.      7
run no.      8
run no.      9
run no.     10
```

```
>top
Select function:
mem
The following tests are available :
memtest
contest
hlctest
Type the first 3 letters to select test.
Select test:
mem
memtest      initiated as mem02
Select function:
>mem02
-- mem 204 test ---- ver 82.12.30 --
testprogram        :a
a: address test
b: bit selection
c: complement test
d: jump test
e: long reliability
f: eprom checksum
Select test:
c
Select function:
list
-- mem 204 test ---- ver 82.12.30 -- LIST OF PARAMETERS :
p  0  testprogram        :      c
p  1  no of runs         :     20
p  2  module no          :    196
p  3  first address (k)  :      0
p  4  first address (w)  :      0
p  6  datacheck          :    yes
p  7  parity check       :    yes
p 10  test size (k)      :      1
p 11  test size (w)      :   1023
p 49  max message        :     10
Select function:
p2:hc6  p49=2  start

-- test area locations  ( hex )  0000  ..  ffff
-- test area size-1              32767
Select function:
run no.      1
run no.      2
run no.      4
run no.      6
**************** run no.     7  : ****************
data error : error in word
module no  :   c6
byte addr. : 2230
expected   :     ffff
received   :     f77f
**************************************************
run no.      8  errors =       1
**************** run no.     8  : ****************
data error : error in word
module no  :   c6
byte addr. : 2230
expected   :     ffff
received   :     f77f
**************************************************
--  Test terminated.
-- mem 204 test ---- ver 82.12.30 -- LIST OF ERRORS :
---------- run no.     9  : ----------
     2 of type :data error : error in word
Total number of errors :     2
--------------------------------------------------
Select function:
>TOP
DISP
Display
TEST  SUB-  RUN   ERRORS     DRIVER-INFORMATION     TEST
      TEST  NO.  DETECTED ACCESSED UNANSWERED LAST STATUS
------------------------------------------------------------
hlc01  a      10      0      4900          0     12 STOPPED ( 3)
mem02  c       9      2       198          0      0 STOPPED ( 3)
Select function:
```

A. REFERENCES

[1] RCSL Nb 52-AA1156:
RC3502 Operating Guide

[2] RCSL Nb 30-M330:
RC3502, TOP80, Installation Guide

[3] RCSL Nb 30-M331:
RC3502, TOP80, Test Operating System, Programming Guide

[4] RCSL Nb 30-M298:
RC3502, VDC Testprogram Package, User's Guide

[5] RCSL Nb 30-M332:
RC3502, MEM Testprogram Package, User's Guide

[6] RCSL Nb 30-M333:
RC3502, CPU Testprogram Package, User's Guide

[7] RCSL Nb 30-M334:
RC3502, COM Testprogram Package, User's Guide

[8] RCSL Nb 30-M335:
RC3502, HLC Testprogram Package, User's Guide

[9] RCSL Nb 30-M336:
RC3502, IMS Testprogram Package, User's Guide

B. REVISION STATUS

Revision 1.0 (81.04.01):

This first edition includes the following restrictions:

- Text written after the NEW-command to "top" is not passed to the test. That should be the case, if it is commands to the test (like commands in TOP at RC8000).

- CHANGE output device as a command to the test is not implemented.

- There is no "timeout"-facility in TOP80, but some tests have a timer (used in communication with hardware). This means that no run-number is displayed every five minuttes. It is only displayed every time 1/10 of the total number of runs is reached.

WARNING:

If a NEW-command results in an initiation of a test which fails immediately then KILL the test. Otherwise "top" may stop when more of this kind of faults have happened. Then you have to restart the whole TOP80 system using the method described in section 4.2.

NO REPLY

This error produces no errormessage. The error is:
You do not get any reply after typing input, which use to send output as reply. To repair: Retype the input-line one or two times.
The error appears, if a a process is given the same name as a removed process, and if the removed process was waiting for input before removal.
Removal is done by either:
"opsys": remove s (Then input to all processes which were existing, will give the error).
"top" : kill/mem01 (Then input to process mem01 will give the error if memtest is NEW'ed again with the same name).

C. PACKAGE CONTENTS DESCRIPTION C.

The supplied 9-track magnetic tape (MTSW2201) has the following files:

File No	Contents	Format
0	label	text
1	package identification	text
2	job to load the package	text
3	programs and jobs	save/load

File No 1 may be inspected in this way:

```
t = set mto mtsw2201 0 1
copy list.yes message.no t
```

File No 2 is a textfile, which contains a job with fp-command to load the package. The package is loaded in this way:

```
t = set mto mtsw2201 0 2
i t
```

Entry Survey

RC3502TOP	The bootfile containing all tests.
AUTOTOP	The job which autload this bootfile.
RC3502CPU	The bootfile containing the basissystem and the stand-alone CPU-test.
AUTOCPU	Autoload job.
RC3502 MEM	The bootfile containing the testsystem and the MEM-test.
AUTOMEM	Autoload job.
RC3502HLC	The bootfile containing the testsystem and COM204-test.
AUTOHLC	Autoloadjob.
RC3502COM	The bootfile containing the testsystem and COM203-test.
AUTOCOM	Autoload job.

RC3502DIAG	The bootfile containing the basissystem and diag-program.
AUTODIAG	Autoload job.
BOOTTES202	The bootfile containing the basissystem and the program for burning your own PROM modules (TES202). Use the bootfiles with names ending on "TES".
AUTOBURN	Autoload job.
TOPTES	The bootfile for burning CPU-test, MEM-test, COM-test, HLC-test, IMS-test, VDC-test and DIAG.
CPUMEMTES	The bootfile for burning CPU-test, MEM-test, IMS-test, and DIAG.
HLCIMSTES	The bootfile for burning HLC-test and IMS-test.
MIRTES	The bootfile for burning the stand-alone VDC mirror.
COMVDCTES	The bootfile for burning the MEM-test, COM-test and VDC-test.
DIAGTES	The bootfile for burning the stand-alone DIAG and TES202 burn program.

RETURN LETTER

Title: RC3502 - TOP35 - Test Operating System, User's Guide RCSI No.: 30-M329

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

42-i 1288

..... Fold here .....

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

Affix
postage
here

REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark