RCSL No: 31-D628

Edition: May 1981

Author: John Munkholm Andersen

BOSS2 INSTALLATION AND MAINTENANCE

A/S REGNECENTRALEN
af 1979

Keywords:

RC8000, RC4000, Basic Software, Operating System, Manual.

Abstract:

This manual explains how a BOSS2 system is generated for a particular installation, some maintenance information is also presented, especially concerning the user catalog, the accounting, and the test output.

(108 printed pages)

Printed by A/S Regnecentralen af 1979, Copenhagen

FOREWORD

First edition: RCSL No 31-D191

This manual explains how a BOSS2 system is generated for a particular installation. The system generation itself is very simple, but some background information is necessary in order to utilize the resources fully.

Som maintenance information is also presented, especially concerning the user catalog, the accounting, and the testoutput.

The manual consists of rather independent chapters written as the need arose and the time allowed. Thus the manual does not pretend to be complete, but we expect it to grow as new chapters are added gradually.

Søren Lauesen
A/S Regnecentralen, July 1972.

Second edition: RCSL No 31-D313

Third edition: RCSL No 31-D421

The manual has now been revised so that it should be possible for unexperienced people to generate a bossversion for a given configuration. Furthermore the manual has been divided into an installation and a maintenance part, each of which has been extended considerably.

Lars Otto Kjær Nielsen and Bent Bæk Jensen
A/S Regnecentralen, February 1977.

Fourth edition: RCSL No 31-D628

Since the third edition of this manual appeared, five new releases of BOSS2 have been issued, and the correction list has grown to 34 pages.

These corrections have been incorporated in the text, which has also been generally revised.

Chapter 5 concerning the accounting system has been rewritten and chapter 9 concerning the test output types has been extended.

The present edition of the manual corresponds to BOSS2 release 17.0 (the Software Package SW8101/1/17.0)

John Munkholm Andersen
A/S REGNECENTRALEN af 1979, May 1981.

TABLE OF CONTENTS PAGE

TABLE OF CONTENTS (continued) PAGE

TABLE OF CONTENTS (continued) PAGE

1. SYSTEM GENERATION 1.

1.1 The Source Text 1.1

BOSS is supplied as a magnetic tape, containing a standard label and 21 textfiles, or as two diskettes containing the 21 text-files.

The files are:

1, bossold — When BOSS is supplied on magnetic tape, this file generates descriptors to the other files on the tape and transfers 'options' and 'jobdescr' to disc. Bossold always generates some service files for managing the BOSS files (chapter 4).

2, options — This file contains options for some standard configuration (section 1.3).

3, jobdescr — This file describes a set of data formats, used during translation of some of the BOSS coroutines.

4, central — Translates into 'bos', i.e. this file is translated by 'i bossbin' into the binary coroutine file 'bos'. Contains code for testoutput, Central Logic, paging, start-up, and initialization.

5, tterm1 — Translates into 'bterm1'. Contains code for common interpretation of on-line commands, for on-line editing, for printing of error messages, and for a few commands.

6, tterm2 — Translates into 'bterm2'. Contains code for most of the on-line commands, for kit change, for request output to the operator, and for job output to the on-line user.

7, tjobstart Translates into 'bjobstart'. Contains code for interpretation of job specification and start-up of a job. Contains code for the loadspecification.

8, tjob Translates into 'bjob'. Contains code for messages from a job (except mount and load messages) and for job termination.

9, tmount Translates into 'bmount'. Contains all code for handling of magnetic tapes.

10, tread Translates into 'bread'. Contains code for handling of paper tape reader, card reader, and other spooled input devices.

11, tprinter Translates into 'bprinter'. Contains code for handling of printers.

12, tprocs Translates into 'bprocs'. Contains code for receipt of messages from unidentified terminals and processes which are not BOSS jobs. Contains various large common procedures for accounting, retrieval of information from the user catalog, book-keeping concerning backing storage resources, and creation of links for remote devices.

13, tbanker Translates into 'bbanker'. Contains code for priority scheduling, resource allocation, swopping, book-keeping of run time. Contains initialization code for catalog cleaning and balancing of main catalog against user catalog.

14, taccount When loaded, this file generates a simple accountjob.

15, tcatupdate Contains the text of the program 'catupdate', which is used to create and update the user catalog. Also contains a text which will create a file 'boptions', containing certain information from 'options'. (chapter 6).

16, tuserout Contains the text of a program 'userout', which may be used to print the contents of the user catalog (see 6.4).

17, testout Contains the text of the external algol procedure 'bossout' and the program 'last', which are used to analyze and print the contents of the test output file (chapter 9).

18, textxref Contains the text of a program which can produce cross references of externals used in BOSS. (For maintenance purposes only).

19, tsaveconv Contains the text of the program 'saveconv' which after a break down can find all the temporary convertfiles which have been produced but not yet printed by BOSS. The files will be transferred to a magnetic tape. The program must be called before BOSS is restarted. The use of the program is described in "BOSS Operator's Manual".

20, tgetconv Contains the text of the program 'getconv' which is used to print magnetic tapes created by 'saveconv'. The use of the program is described in "BOSS Operator's Manual".

21, tusercat Contains a job which creates a simple user catalog.

1.2 Translation of the Source Text 1.2

Notice that backing storage files will be created with scope 'user'. This makes it possible to have more versions of the system at the same time. It may be convenient to have the normal version on system scope while changes in options etc. may be debugged in a local version.
The following three examples show how the system is generated, depending on the document used for storage of the source text.

1.2.1 Source Text on Magnetic Tape


```
-> bossold=set <modekind> <tapename> 0 1

-> i bossold    ; the files 'options' and 'jobdescr' are trans-
                ; ferred to disc, file descriptors to the rest of
                ; the files are created, and the service files are
                ; generated (see chapter 4).
-> i trf        ; modification of standard options (see 1.3.3).

-> i bossbin    ; translation of the BOSS coroutines (see 4.1).

* att ->s
-> remove all boss run

-> bosstest=set 200 disc ; creation of testoutput area
-> scope user bosstest

-> i taccount   ; creation of an account job (see chapter 5).

-> i tusercat   ; creation of an experimental usercatalog with
                ; contents as shown in the example in chapter 6.
```

1.2.2 Source Text on Backing Storage


```
-> i bossold    ; generation of service files (see chapter 4).
                ; As (in this case) the file descriptors to
                ; the rest of the files are already present,
                ; the following message will appear:

file descriptors left unchanged
-> i trf        ; modification of standard options (see 1.3.3).

-> i bossbin    ; translation of the BOSS coroutines (see 4.1).

* att -> s
-> remove all boss run
```

```
-> bosstest=set 200 disc ; creation of
-> scope user bosstest   ; testoutput area

-> i taccount  ; creation of an accountjob (see chapter 5)

-> i tusercat  ; creation of an experimental usercatalog with the
               ; contents of the example in chapter 6.
```

1.2.3 Source Text on Diskettes 1.2.3

```
-> fdload s18101.1 ; the files 1-13 are loaded to the system disc
                   ; with scope 'temp', i.e. when BOSS is started
                   ; up, they will be removed.
-> fdload s28101.1 ; the files 14-21 are loaded as above.
-> i bossold       ; generation of service files (see chapter
                   ; 4).
-> i trf           ; modification of standard options (see
                   ; 1.3.3).
-> i bossbin       ; translation of the BOSS coroutines (see
                   ; 4.1).

 * att -> s
-> remove all boss run
   ready

   to boss
-> bosstest = set 200 disc  ; creation of
-> scope user bosstest      ; testoutput area
-> i taccount               ; creation of accountjob
-> i tusercat               ; creation of an experimental user
                            ; catalog
```

1.3 Options 1.3

When translating the BOSS coroutines all information used for trimming BOSS must be listed in a file, named 'options'. To ease the system generation some standard configuration is described in file No 2 (options). The trimming is done by editing this file.

1.3.1 Options to Consider

Normally only the first 3 pages of the options need to be considered:

Page 1 defines the computer configuration on which BOSS is to run. For instance, various device numbers are stated here.

Page 2 defines the main parameters, for instance, the maximum number of jobs which may be handled simultaneously, the amount of login resources to each terminal, the maximum size of an off-line job file.

Page 3 defines the time classes and the class margin. Further, standard and maximum values of options. Notice, that the class margin ought to comprise resources for at least one job with standard resources. Otherwise, short jobs with standard resources cannot be guaranteed to bypass longer jobs. For some installations a class margin corresponding to two standard jobs may be advantageous.

In the following table the most important options concerning various elements and possibilities of BOSS are listed.

Main console, terminals

...1... i1, i4
...2... i45, i44, i48, i47, i113, i114
...4... i7, i9, i161, i16, i175, i176
...5... i3, i179, i181
...5a.. i177

Operator display

...1... i171
...4... i172, i173
...5a.. i174

Printers, remote printers

...1... e23-e17, e35-e36, e58-e59, i190
...2... i67, i104, e40, e47
...3... i135, i147
...4... i115, i191, i192
...5a.. i72, i77, i79, i73, i74, i167, i168,
i169, i170, i180, i75, i185, e38-e39, i68

Paper tape reader, special reader device (control), remote readers

...1... e51-e55, e55-e52, e53, e23-e17
...2... i41, i117, i119, i162
...3... i146
...4... i32, i193
...5... (i42)

Punched card reader, remote card readers

...1... e54, i93, i200, e23-e17
...2... i203, i52, i118, i163, i201
...5... (i53)

Magnetic tape stations

...1... e24-e25
...2... i35
...3... i106, i132, i138, i144, i145, i150
...4... i31, i33

Simulated conversational devices

...1... e18-e19, e20-e21
...4... i9, i16

Flexible disc drives (diskette stations)

...1... e25-e28, e60-e61

Other devices not simulated by BOSS

...1... e23-e28

Devices which require operator attention before use

...1... e25-e17

Drum/disc

(See 1.3.2)

Other backing storage devices

(see 1.3.3)

Accounting

...2... i105
...3... i136, i148
...5... e70, e71, i14

Miscellaneous

...1... e0, e79, e60-61
...2... i116
...5... i11, i178, (e6, e7, e9), e16
...7... (e99, e100), e101

1.3.2 Options Concerning System Disc 1.3.2

A number of options concern resources on the system disc (i.e. the disc with the name 'disc'). These options (with their standard values) are:

```
options page 2:
i47 = 280        ; max. no. of segments in a basis file
i105 = 24        ; no. of segments in account file
i114 = 25        ; login slices for each terminal
i162 = 4         ; max. length of job file for a papertape job
                 ; (slices)
i163 = 4         ; max. length of job file for a cardjob (slices)

options page 3:
i108 = 35        ; temp disc slices in class margin
i121 = 25        ; temp disc slices, std. value for jobs
i137 = 4         ; std. value for output (slices)
i140 = 16384     ; std. value for jobsize (halfwords)
```

The reason for the inclusion of i140 (jobsize) here, is that it determines the size of the swoparea for a standard job.

The standard values are reasonable if the slicelength of the system disc is 8.

The following table gives suggested values of these options for the slicelengths normally found on RC8000 system discs.

DISC	slice-length (segm)	i47	i105	i114	i162	i163	i108	i121	i137	i140
RC8221	7	210	84	38	4	4	42	30	8	13312
RC8222	14	280	84	24	2	2	26	20	4	13312
RC8223	21	315	84	18	2	2	20	15	3	13312
RC8224	42	420	84	12	1	1	13	10	2	13312
RC8225	84	504	84	7	1	1	10	8	1	13312
RC8226	168	672	168	5	1	1	8	6	1	13312

In connection with these changes you should also change the options concerning the number of catalog entries for a standard job (options page 3) to: i107 = 12 and: i126 = 10, as the standard values given are unnecessarily high.

If you want to use other values of these options, you should chose them so that:

1. i108 = i121 + i137 + swoparea (slices)

   The size of the swoparea may be calculated from the value of i140. (1 segment = 512 halfwords)

2. i114 = max. basis file (slices) + i137

The values in the table give the following standard values of the job specifications:

DISC	slicelength (segments)	temp disc (segments)	output (characters)
RC8221	7	210	43008
RC8222	14	280	43008
RC8223	21	315	48384
RC8224	42	420	64512
RC8225	84	672	64512
RC8226	168	1008	129024

1.3.3 Other Discs

1.3.3

Discpacks (or logical backing storages) on which users are given permanent resources in the user catalog, may only be mounted on the disc drives mentioned in the list in options page 1 after e22. Therefore you should normally include all disc drives, except the one for the system disc, in this list.

The option i29 (options page 1) stating total number of drum and disc devices, must be exactly equal to the number of disc drives (logical backing storages) stated in the monitor. Otherwise BOSS refuses to start up.

The option i30 (options page 2) stating the maximum number of private discpacks in one project may as a useful standard be equal to: (i29 - 1)*2.

my options
udshoft

1.4 Standard Options

The following is a listing of the standard option file.

```
; rc4000 rialto   20.3.72                boss 2, options ...1...

; definition of computer configuration.

; 80 06 28, 57  ; version id:

b. i220, e110, r199  w.
e0: <:rialto:>; name of installation (may also contain identification of version)
    0,0
e29=k-e0      ; length of text
e79=3     ;    monitor release
e80=-1    ;    target mashine:          -1 means rc4000
          ;                              1 means rc8000
i1=2      ;  device number of main console.
i171=-1   ;  device number of operator display (-1 if not present)
i4=9      ;  number of terminals which may be logged in
          ;  simultaneously. excluding main console.
i15=1014  ;  max number of slices on any exchangeable disc kit. (must be even)
i27=1     ;  drum exists (1) or absent (-1). the main catalog must
          ;  be on drum if it exists. disc must always be present.
i29=3     ;  total number of drum and disc devices.

e24: h.
6, 7, 8   ;  list of device numbers for standard tape stations.
e26=k-e24
10        ;  list of device numbers for special tape stations.
e25=k-e24
          ;  list of device numbers for remote process control
e28=k-e24 ;  devices not simulated by boss
e18:      ;  list of device numbers for conversational process
          ; control devices.
e19=k-e18 ;
e51:
0         ;    list of device numbers for std tape readers
e55=k-e51 ;    number of std tape readers
          ;    list of device numbers for special reader devices
e52=k-e51 ;    number of std readers + special input devices
e53=0     ;    number of remote batch tape readers
e22:      ;  list of device numbers for exchangeable disc drives
14        ;  which may be reserved by jobs
e23=k-e22 ;
          ;  list of device numbers for special devices without
          ;  a mnemonic name
-1        ;  device number of plotter, -1 if plotter is unavailable.
1         ;  device number of punch, -1 if punch is unavailable.
-1        ;  job controlled cards allowed (1) or forbidden (-1).
1         ;  job controlled printer allowed (1) or forbidden (-1).
1         ;  job controlled reader allowed (1) or forbidden(-1).
e17=k-e24, w.

e44:      ;  list of process names simulated by boss (8 bytes each)
e45:      ;    end of list of process names

e20: h.   ;  list of buffer sizes (bytes) for process control
e21=k-e20,w.;  devices.  must match the list from e18 and e44.
          ;    (all buffer sizes must be even)

e35:
5         ;  list of device numbers for line printers.
e36:
i71=e36>1-e35>1;  (number of paper printers)
e58: h.   ;    list of preselected papertypes (1 byte for each printer)
-1, r.i71 ;    -1 means all papertypes allowed on corresponding printer
e59: w.   ;    the list must correspond to the list of printers (option e35 - e36)
i190=0    ;  number of remote batch printers

e54: h.
          ;  list of device numbers for std card readers
i93=k-e54 ;  number of std card readers
e60:      ; list of devicenumbers for which the call-command are allowed
          ; normally the devices should also be in one of the lists e25, e23
e61=k-e60 ; number of call-devices
w.
i200=0    ;  number of remote batch card readers
```

```
; rc4000 rialto   20.3.72                boss 2, options ...2...

; main parameters. should be considered at each installation.


i35=10     ;  total number of suspend buffers.
i45=20     ;  number of pseudo jobs including the account job. every
           ;  terminal has a pseudo job. the remaining pseudo jobs
           ;  are available for batch jobs and internal jobs. i45
           ;  may not exceed 59.
i203=i93+i200; number of temporary card job files.
             ;   (boss uses a temporary job file while examining the
             ;    job specification, i.e. this option gives the number
             ;    of simultaneous job preparations)
             ;    0 <= i203 <= i93+i200
i206=e52+e53 ; number of temporary tape job files (see above);

i44=5      ;  number of correction segments to each basis file. about
           ;  12 corrections may be stored on one segment.
i48=80     ;  max number of corrections to each basis file. should be
           ;  slightly larger than 12*i44.
i47=280    ;  max number of segments in a basis file. if i48 + i47
           ;  <104, the job file page will be 1 segment.
           ;  if i48 + i47<360, the job file page will be 2 segments.

i41=30     ;  number of segments in disc area for spooling of job
           ;  controlled paper tapes.
i52=30     ;  number of segments in disc area for spooling of job
           ;  controlled card input.
i67=80     ;  number of segments in disc area for spooling of job
           ;  controlled printer (must be greater than one).

i104=40    ;  total number of convert buffers.
i105=24    ;  number of segments for account file. one segment
           ;  contains 16 account records. i105 determines how often
           ;  the account job is executed. (must be a multiplum of the slicelength on disc)
i113=6     ;  login entries for each terminal.
i114=25    ;  login slices for each terminal.
i116=30    ;  minimum number of segment places in core store for
           ;  boss pages. determines the largest size of a job and
           ;  the combined size of two jobs in case they may reside
           ;  in core simultaneously.
i117=0     ;  standard mode for job tapes. ( 0= flexo, 1= iso).
i118=10    ;  standard mode for reading card jobfiles.
i119=0     ;  no of remote terminal stations which may requesx
           ;  loading of job tapes simultaneously
i201=1     ;  release rb card readers:
           ;    i201= -1 :  only release on malfunction etc.
           ;    i201=  1 :  release also on card reader ready.
           ;    (if number of possible rb cardreaders exceeds i200,
           ;     then i201 should be 1, thereby giving the other
           ;     cardreaders a change of loading a batch of cards).

i162=4     ;  max length of job file for a paper tape job (slices).
i163=4     ;  max length of job file for a card job (slices).
e40=600    ;  length of printer buffer in core store (bytes). the
           ;  larger the buffer, the less the time spent in printer
           ;  administration. (must be even)
e47=512    ;  length of remote batch printer buffer in core store (bytes). (must be even)

i30=4      ;  max number of private disc kits in one project.
           ;  has effect on the size of the job descr page and on the
           ;  possible contents of the user catalog.
i20=0      ;  max number of simultaneously named telex devices which a job
           ;  or a terminal user can handle.  has effect on the size of
           ;  the job descr page.
```

```
; rc4000 rialto   20.3.72                boss 2, options ...3...

; time classes, class margin, common standard values and max values
; for options.

i101=30   ;  definition of class 0 jobs: gross run time<i101 seconds
i102=3*60 ;       -    -    -  1   -          -      <i102   -
i103=18*60;       -    -    -  2   -          -      <i103   -

; number of resources in class margin:

i106=0    ;  standard tape stations
i107=20   ;  temporary catalog entries (includes one for primout and
          ;  one for swop area).
i108=33   ;  temporary disc slices (includes room for primout and
          ;  swop area).
i109=5    ;  message buffers (including one to boss for receiving
          ;  messages from jobs)
i110=8    ;  area processes (includes one for swop area).
c. i27
i112=30   ;  temporary drum slices.
z.
c.-i27
i112=0    ;    if no drum then i112 is set to 0
z.

; common standard values for options (may be redefined by user
; catalog and job specification):
i122=2.111111100000 ;    bits defining std terminal user rights

i121=25   ;  temp disc (slices)
i126=18   ;  temporary catalog entries
c. i27
i127=20   ;  temp drum (slices)
z.
c.-i27
i127=0    ;    if no drum then i127 is set to 0
z.
i132=0    ;  stations
i135=8    ;  cbuf
i136=0    ;  accounts
i137=4    ;  output (slices, 768 characters to a segment)
i138=0    ;  suspends
i139=75   ;  time (units of 0.8 seconds)
i140=16384;  size
i141=4    ;  buf
i142=7    ;  area
i134=0    ;  waiting time deliberately swopped out (seconds)
i131=0    ;  minimal information: 0==no, 1==yes.
i123=0    ;  number of simultaneous used rb-devices

; common max values for options (may be redefined by user catalog).
; further restrictions from start-up resources and class margin.
; the values may be very high as a useful standard.

i143=6    ;  key
i144=10   ;  stations
i145=100  ;  mounts
i146=1000 ;  tapes
i147=100  ;  cbuf
i148=500  ;  accounts
i149=200  ;  output (slices)
i150=100  ;  suspends
i151=24*60*75; time (units of 0.8 seconds)
i152=100000  ; size
i153=100  ;  buf
i154=100  ;  area
i155=10   ;  internals
i156=24*60*60; latest (lower limit, seconds)
i157=60*60;  waiting time deliberately swopped out (seconds)
i124=120  ;  max number of simultaneous used rb-devices
          ;  (the no. of entries in the device table)
```

```
; rc4000 rialto   20.3.72                boss 2, options ...4...

; various time constants. need not be considered in the first
; approximations.

i5=2        ;  time from provoke break to expiration of run time.
            ;  (seconds)
i6=750      ;  jobs with a gross run time > i6 will not be terminated
            ;  automatically when run time expires (units of 0.8 sec).
i7=375      ;  time interval between successive warnings to operator
            ;  when a long job exceeds its run time (units of 0.8
            ;  seconds).
i9=3        ;  conversatinal jobs run with high priority
            ;    during i9 * 0.8 seconds after each input
            ;    demand. after that normal priority is resumed.
i31=60      ;  time for operator to mount one mag tape (seconds).
i32=20      ;    -   -       -     -   -    -  paper tape (seconds).
i33=2       ;  time interval in station scan (seconds). the scan only
            ;  takes place when unidentified tapes are mounted on a
            ;  tape station which cannot be a job resource (e.g.
            ;  station for test output or spare station).
            ;  a mounted tape will always be recognized by boss
            ;  within the next i33 seconds.
i115=5*75   ;  gross run time for printing and converting. may be used
            ;  to adjust the matching between expected finishing time
            ;  and real finishing time in case disc room is the usual
            ;  bottle neck.
i161=375    ;  batch jobs and internal jobs with a gross run time >
            ;  i161 will cause a warning message to the operator when
            ;  they are enrolled. (units of 0.8 seconds)
i16=18000   ; gross run time for conversational jobs(conversation
            ; with terminal or process control device). units of 0.8 sec.
i172=2      ;  time between operator warning when pending requests (via operator display)
i173=20     ;  time between refreshing of operator display
            ;    (when i172 > i173 : no operator warnings are made)
            ; transmit:
i175=10     ;   maximum waiting time allowed between the message >start transmit<
            ;   and the actual start of the tape reader. (unit of appr. 2 sec.)
i176=1      ;   maximum time the tape reader may be stopped,
            ;   without resulting in >end transmit<. (unit of appr. 2 sec.)
            ; rb printers:
i191=1      ;   max waiting time when reserving rbprinter (minutes).
            ;     (if the remote batch terminal has not granted the
            ;      reservation within i191 minutes, boss will postpone
            ;      the convert, and proceed to the next conversion.
            ;      when boss again reaches the same convert, boss will
            ;      try again to reserve the printer etc.)
            ;      0 <= i191 <= 127, i191=127 means 'wait till you get it'
i192=60     ;   time interval in scan of rbprinter convert queue (seconds).
            ;     (the scan only takes place in case another process,
            ;      running in parallel with boss, uses some of the rbprinters
            ;      needed for converting of files)
i193=4500   ;  arrival is set to now-i193 when input arrives to a job using the general
            ;  input spool mechanism (control <device>). units of 0.8 seconds
```

```
; rc4000 rialto   20.3.72                boss 2, options ...5...

; various options. need not be considered in the first
; approximations.

i3=100    ;  length of terminal buffer in core store. the larger the
          ;  buffer, the less the time spent in administration of
          ;  terminal output. at least one line (bytes). (must be even)
i11=80    ;  search fraction (pct). the priority algorithm for pages
          ;  tries to keep the search length for free core close to
          ;  i11.
e70=0     ;  accountrecord at jobstart
          ;   (e70=0: no, e70=1: yes)
e71=0     ;  userindex in accountrecords
          ;   (e71=0: no, e71=1: yes)
i14=2     ;  max number of account records produced by account job.
          ;  as a standard 2, one for the run and one for primary
          ;  output.
i42=2     ;  length of reader buffer for job tapes and loads
          ;  (segments).
i53=2     ;  length of card buffer for job files and loads (segments).
i178=0    ;  size max allowed or not in job specification
          ;   (i178=0 : not allowed, i178=1 : allowed)
i179=0    ;  main console to be considered a normal user console
          ;    (i179=0: no, i179=1: yes)
          ;     warning: should not be set to 1, since job output
          ;              and requestlines etc. will be mixed on
          ;              the console, which will make it difficult
          ;              for the main operator.
i181=0    ;  reservation of terminals
          ;  0: all terminals exept main console are reserved at login
          ;  1: as  0 and the att command is not allowed
          ; -1: no terminals are reserved at login
          ; -2: as -1 and the att-command is not allowed
e6=-1     ;  change testoutput tape. (e6=1 means included, e6=-1 means not included)
e7=-1     ;  testoutput and initialization constants printed by
          ;  means of fp (1) or not printed (-1).
e9=7      ;  initial test mode. a bit pattern specifying the test
          ;  output to be produced. bit = 1 means produce output.
e16=1     ;  e16=1 means: boss is not allowed to run in monitor mode.
          ;               if boss is started in monitor mode then the protection
          ;               key of boss core area will be changed during start up.
          ;               in this case boss will always use one protection key.
          ;  e16=-1 means: boss is allowed to run in monitor mode.

; additional differentiated test patterns:

e81= 7    ; timer
e82= 7    ; banker
e83= 7    ; unknown sender
e85= 7    ; cardreader
e86= 7    ; reader
e87= 0    ; watch dog
e88= 7    ; request display
e89= 7    ; pager
e90= 0    ; operator display
e91= 7    ; kit changers
e92= 7    ; printers
e93= 7    ; rewinders
e94= 7    ; remoters
e95= 7    ; commandios
e96= 7    ; psjobs
e97= 0    ; termouts
e98= 7    ; rb cardreaders
```

```
; rc4000 rialto    4.11.74                boss 2, options  ...5a...


i72=10     ;  max number of printer pages which may be repeated.
i77=20     ;  max number of segments which may be repeated in job
           ;  controlled printing.
c. i77-i67, i77=i67-1 z.
i79=2      ;  standard number of pages repeated.
i73=5      ;  number of lines assumed for a VT-character.
i74=65     ;  number of lines assumed for a printer page. i73 and i74
           ;  define the page boundaries used with the repeat- and
           ;  kill- command when no FF is present.
i167=36    ;  size of printer triangle in head of a print file (lines).
i168=0     ;  number of FF preceding first page of a print file.
i169=0     ;    -    -   - succeeding last  -   -  -   -     -
i170=2     ;  min no of pages before kill of printer output
i180=3     ;  number of errors allowed in remote printing
           ;  before conversion is transferred to the standard printer
i75=0      ;  destroy contents of non-permanent convert files
           ;    after printing (i75=0: no, i75=1: yes).
           ;  standardvalue for mode in printer output:
           ;  0=normal mode, 2=doublebuffer mode.
           ;  mode 2 is not allowed if monitor release < 5
c. e79-5
i185=2     ;  if monitor release >= 5 then mode:=2;
z.
c. 4-e79
i185=0     ;  if monitor release <  5 then mode:=0;
z.
i99=2      ;  number of tape marks written after boss label
i174=25    ;  number of lines on operator display
i177=2000  ;  command input: maximum number of timer interrupt of input
           ;    from a terminal, when typing one line. if the limit is
           ;    exceeded, the terminal will be logged out automatically.
           ;    the effect of i177 will be dependent on the monitor options
           ;    for each terminal, but as a standard a timer interrupt
           ;    will occur, if no characters have been typed for approx. 10 secs.

e30=512    ;  length of file buffer (bytes) in on-line editor. at least
           ;  512.
e38:       ;  list of paper type names. used in output to operator.
<:monitor 1<0>:>  ;  each name must terminate with a null-character
<:a4 høj 1<0>:>   ;  and may not exceed 11 characters.
<:a4 tv 1<0>:>
<:monitor 2<0>:>
<:a4 høj 2<0>:>
<:a4 tv 2<0>:>
<:monitor 3<0>:>
<:a4 høj 3<0>:>
<:a4 tv 3<0>:>
e39:
i68=10     ;  number of paper names + 1.
```

```
; rc4000 rialto 20.3.72                boss 2, options ...6...

; pages placed on drum (0) or disc (1). these options effect the
; time spent in page transfers at run time. the options have no
; effect when drum only or disc only is present. the options need
; not be considered in the first approximation.

112=1     ; account buffer
113=1     ; account code
117=1     ; kit change
121=1     ; request lines
122=1     ; request display code
123=0     ; terminal output code
125=0     ; terminal output buffer
128=0     ; bs adjust code
137=1     ; lookup usercat code (initcat)
138=1     ; usercat buffer
139=1     ; spool area for job controlled paper tapes
140=0     ; reader buffer for job tapes and loads
143=0     ; reader code
150=1     ; spool area for job controlled cards
151=0     ; card reader buffer for job files and loads
154=0     ; card reader code
156=0     ; commandio code, list
157=0     ; job file page
158=0     ; buffer for basis file
159=0     ; buffer for primary output
160=1     ; correction segments
161=1     ; unknown sender code
163=0     ; code for login, logout, get, save, etc.
164=1     ; code for remote batch communication
165=0     ; paper printer code
166=1     ; spool area for job controlled printer
170=0     ; paper printer buffer
180=0     ; pseudo job code, start and finis
181=0     ; job description page
182=1     ; magnetic tape description
183=1     ; tape mounting code, job part
184=1     ;  -       -      - , remoter part
185=1     ;  -       -      - , rewinder part
186=1     ;  -       -      - , watch dog part
187=1     ;  -       -      - , label read-write
188=1     ;  -       -      - , rewinder que
189=1     ;  -       -      - , remoter que
190=1     ;  -       -      - , name, label command
198=1     ;  -       -      - , mount action table
191=0     ; pseudo job code, parent messages
192=0     ;    -     -    - , primary input/output
194=0     ;    -     -    - , job controlled paper tape
195=0     ;    -     -    - , -       -     cards
196=1     ;    -     -    - , load commands
197=1     ; display and request command
1100=0    ; banker and timer code and variables
1164=1    ; convert buffers
1165=1    ; paper description page
1166=0    ; printer variables
1202=1    ; card reader variables
1204=1    ; snap code-page
1205=-1
```

```
; rc4000 rialto   20.3.72                boss 2, options ...7...

; maintenance parameters. should only be changed as a consequence
; of code modifications.

; monitor dependent variables:
; should only be changed as a consequence of monitor code changes
e48=0     ;  relative in remote batch process description: creation number
e49=18    ;  relative in area process description: number of segments
e50=26    ;  relative in disc process description: slice length
;       rc4000:              rc8000:
c.-e80, e56=14, z., c.e80, e56=16, z.;  length of register dump area;
c.-e80, e57=38, z., c.e80, e57=80, z.;  relative in process descr. first dumped register;

i2=i45+e25+1+i93+i71+1+1  ;  number of request lines:   psjobs +
; stations + reader + card reader + printers + deadlock + printline
i2=i2+i190+i200+1; +remote batch printers and -cardreaders + hotnews

i8=17     ;  net time between warnings to operator
i18=4     ;  segments in pooltable page (tterm2, kit).
          ;  each segment holds approx. 50 pool descriptions;
i19=10    ;  length of pool description (halfwords, must be even)
i46=100+(:i45+2+1:)/2*2;  work space on job file page,
c. 116-i46, i46=116 z. ;    (halfwords, must be even).
i49=5     ;  max number of parameters to an option
i62=i44   ;  segments in save and clean cat buf
c. i44-i62-1, i62=i44 z. ; (at least 144 segments
c. (:i15+34:)>9 - i62, i62=(:i15+34+511:)>9 z. ; and at least i15 + 34 bytes)
i78=168   ;  number of paper queues

i129=0    ;  login slices drum
i133=1    ;  internals in class margin

e1=400    ;  max number of segments in drum core (initialization table)
e2=400    ;   -   -   -     -   - disc  -         -
e3=15000  ;  max size of corutine file (part output by slang only). (must be even)
e4=1200   ;  length of external table (bytes). (must be even)
e5=100    ;  length of pre-externals (bytes). (must be even)
e8=1      ;  test output of pages. e8=1 abs addresses, e8=-1 virtual
          ;  addresses.
e10=72    ;  length of request lines (bytes). (must be even)
e46=512   ;  size (bytes) of one request line page (must be a multiplum of 512)
e12=16    ;  length of corutine description (bytes)  (must be even)
e13=6     ;  length of semafore description (bytes)  (must be even)
e11=512   ;  size of job file of operator display (at least= 2 + external 178) (must be even)
c. i27
e99 : <:drum:>, 0, 0 ; name of bs-device for drumcore (the name must occupy 4 words).
z.
c.-i27
e99:  <:disc:>, 0, 0 ;
z.
e100: <:disc:>, 0, 0 ;  -   -  -    -   - disccore ( -   -    -     -   -  -   ).
e101: <:boss2 operating system  :>,0 ,0  ; text appearing after login
                                         ; (the text must occupy 10 words)
```

```
; rc4000 rialto   4.11.74                    boss 2, options  ...7a...

; exact number of corutines:
e14=i4+1 +145     ;  commandios+pseudo jobs
e14=e14+2+i4+1    ;  +unknown sender+pager+termout+request display
e14=e14+1         ;  +main console termout
e14=e14+2+e52+i93+i71;+banker+timer+readers+card+printer
e14=e14+1190      ;  +remote batch printers
e14=e14+e53       ;  +remote batch readers
e14=e14+1200      ;  +remote batch card readers
e14=e14+e25+e25   ;  +rewinders+remoters
e14=e14+e23       ;  +kit change
c. e25-1          ;  +watch dog
e14=e14+1  z.     ;
c. i171
e14=e14+1, z.     ;  + operator display

; exact number of senders:
e15=i4+2+145      ;  commandios+unknown sender+pseudo jobs
c. i171
e15=e15+1, z.     ;  + operator display

e27=e25+i35+145   ;  number of entries in tape table:
                  ;     mounted tapes+suspended works+wanted tapes
e31=i113*(:14+1:)+2+1203;  preoccupied entries:  login, account spare, job
                  ;     tape, card job
e32=i114*(:14+1:)+1+i162+i163*1203; preoccupied disc slices:  login, account
                  ;     spare, job tape, card job
e33=i4+1+1+145+1+1;  preoccupied mess buf's:  commios, banker, pseudo
e33=e33+1+e23     ;     jobs, timer, pager, unknown, kit changers
e33=e33+e25+1+e52+i71+i93; mounters, watchdog, readers, printers, cards
e33=e33+i190      ;     +remote batch printers
e33=e33+e53       ;  +remote batch readers
e33=e33+1200      ;     +remote batch card readers
c. i171
e33=e33+1, z.     ;     +operator display
e34=i4+1+1+145+1  ;  preoccupied area procs:  commios, save, pseudo
e34=e34+e23       ;     jobs, unknown, kit changers
e34=e34+i71+1+i93;     printers,  reader, cards
e34=e34+i190      ;     +remote batch printers
e34=e34+1200      ;     +remote batch card readers
```

```
; rc4000 rialto   21.1.76      boss 2, options   ...7b...

; common variables for all codepages:

e37=2           ;  length of printer entry
e41=72          ;  length of convert buf
e42=8           ;  length of printer point description
e43=12          ;  length of printer operation


r99=22          ;    size of host operation

;  lookup (and reserve) device                                  r0-r9

;  call:

                    r0 = 4       ;  abs ref message area
                    r1 = 6       ;  name
                    r2 = 14      ;  host ident
                    r3 = 18      ;  mode
                    r4 = 19      ;  kind
                    r5 = 20      ;  timeout
                    r6 = 22      ;  catalog base

;  error return:
                    r100= 4      ;  result

;  normal return:
                    r101= 4      ;  0 (name table address of device)
                    r102= 6      ;  (rc device no)
                    r103= 10     ;  rc mode
                    r104= 8      ;  rc kind
                    r106= 12     ;  max blocksize


;  lookup host:                                  r40 - r49

;  call:
                    r40 = r0     ;    abs ref message area
                    r41 = r1     ;    process description

;  error return:          no parameters

;  normal return.
                    r142=r102    ;    host description


m. rc4000 rialto   20.3.72  boss 2 options
i.
```

1.5 Editing the Options - An Example 1.5

As mentioned in section 1.3 the trimming for the actual installation is performed by editing the standard options supplied. In subsections 1.2.1 and 1.2.2 this editing is assumed to be written on a paper tape (trf = flexomode). The following example shows how it may look.

NB: If the source text is on disc, the standard options are removed during this editing.

```
message   edit of standard options
(optt    = edit options
if ok. no
end)

1./...1.../,
1./e0:/, r/rialto/RC COMPUTER,1981.02.18/, ; testoutput ident.
1./e80=/, r/=-1/=1/,                       ; rc8000
1./i15=1014/, r/=1014/=2046/,              ; max no of slices
1./i27=/, r/=1/=-1/,                       ; no drum
1./i29=1, r/=3/=2/,                        ; backing storage devices
1./e24:/, l1, r/6, 7, 8/10, 11/,           ; std stations
1./e26=/, l1, r/10/ /,                     ; no special stations
1./e35:/, l1, r/5 ;/5, 42, 43 ;/,          ; std printers
1./e58:/, l1, d,                           ; preselected papers:
i/
-1 ; printer 5: all except paper 7 and 8
7  ; printer 42: only paper 7
8  ; printer 43: only paper 8
/,

1./...3.../,
1./i140=/, r/=16384/=13312/,               ; std size

1./...7.../,
1./e101:/, d,
i/
e101: <:RC COMPUTER:> 0, 0, 0, 0, 0, 0     ; login message
/,
```

```
f
scope user optt
if ok.no
end
clear user options
rename optt. options
end
```

2. START-UP 2.

After system generation BOSS is started by the following commands:

```
att s
all boss run          ; the BOSS-process must be created with
                      ; system bases

base abs <min> <max>  ; if the system to be started is created on
                      ; system scope, this command may be omitted;
                      ; otherwise it is used to set the catalog
                      ; base of BOSS according to the base of the
                      ; system

bos                   ;
```

The file named 'bos' contains a two-pass loader, which loads the 9 other coroutine files one by one. Each file is entered and allowed to initialize itself. The files reserve room and define addresses in the virtual store and in primary storage. The address definitions are entered in a table of external symbols.

In the second loader pass, the files are loaded and entered again exactly as in the first pass, but now the external symbols are finished and addresses become complete. Certain functions which were dummy in the first pass are active now. For instance, the files now move code, tables, and variables to the virtual store.

At the end of the second pass, BOSS scans the catalog and tries to clear all temporary and login files. At the same time, the actual permanent resources for each user and project are determined. Finally, the user catalog is scanned and all permanent rest claims are updated according to the actual resources used. When a user has run under s, for instance, it may happen that the actual files exceed the claims stated in the user catalog. In this case BOSS just leaves a rest claim of zero, so that the user cannot get more permanent resources under BOSS. This balancing of

user catalog against main catalog involves all bs-devices mounted for the moment. Entries in the user catalog referring to other disc packs are left unchanged.

During the initialization, a well defined access to various devices and files is established. For instance, all magnetic tapes available for jobs are unloaded.

2.1 Warnings During Start Up 2.1

At start up BOSS may write some warnings on the main console, concerning catalog balancing, testoutput, and accounting.

1. project no: <integer> rest claim: <integer> {slices / entries} on <document>

This message occurs when a project has used more permanent resources on the specified document than allowed according to the user catalog.

The message may be caused by:

a) creation of files under 's'
b) decrease of claims in the user catalog
c) overlap of intervals in the user catalog.

2. *** test output inactive

Occurs if no area exists and no tape is mounted with the name 'bosstest', or if 'bosstest' is not created with system scope.

3. <error description>
   accountjob not ok

Occurs if no jobfile with the name 'accountjob' exists with system scope, or if it is not a legal job.

2.2 Error Messages During Start Up 2.2

If BOSS gets into serious troubles during startup, it writes an error message on the main console and then stops (with BOSS fault 5). The possible error messages are:

1. coroutine area missing
   one of the binary coroutine files is not visible from BOSS' catalog base.

2. create disccore error
   during start up BOSS creates two areas for its virtual core (disccore and drumcore). If the creation of disccore is unsuccessful this message appears.

3. create drumcore error
   see 2.

4. dummy answer coroutine
   one of the binary coroutine files cannot be used.

5. external outside limits
   references are exchanged between files via externals; such an external is out of range.

6. input error virtual core
   a work file could not be read during start up.

7. i116 too small : w0=min value
   the number of segment places in core is too small.

8. max coroutine too small
   option e3 is too small, there is not room for one of the coroutine files.

9. output error virtual core
   during start up writing in a work file was unsuccessful.

11. table reservation error
    one of the tables, which can be seen in section 3.1 is created with wrong size, possibly because different options have been used for translating different coroutine files.

12. transfer error coroutine
    BOSS cannot get started because of input troubles when reading a coroutine file.

13. virt core table too small
    possibly same cause as for error No 11.

14. too many bs devices
    appears if the installation (the monitor) is equipped with more bs devices than described in BOSS' options (i29).

15. catalog input trouble
    the main catalog could not be accessed, probably because of hardware errors.

16. drum or disc missing
    the value of the option i29 is too high.

17. entries do not match usercat
    the user catalog is inconsistent, most likely because the latest translation of it was unsuccesful. Retranslate and check the output from the translation carefully.

18. usercat incorrect
    the user catalog is inconsistent. See 17.

19. user cat input trouble
    the user catalog could not be accessed, either because of hardware errors, because it did not exist, had wrong scope, or was reserved by another process.

20. virtual storage too big
    the total demand on virtual storage exceeds 4095 segments. Check the values of the options i41, i52, and i67 (options page 2).

21. time inconsistence accountf1
    date and time now (as defined by the system clock) is earlier than date and time when BOSS was last running.
    The first part of chapter 5 describes how the situation may remedied.

During start up the banker is called to investigate whether the resources promised by the BOSS options and the resources stated in the user catalog are in accordance with the resources given by the monitor to the BOSS process. If the comparison of resources is not ok, one of the following messages may appear:

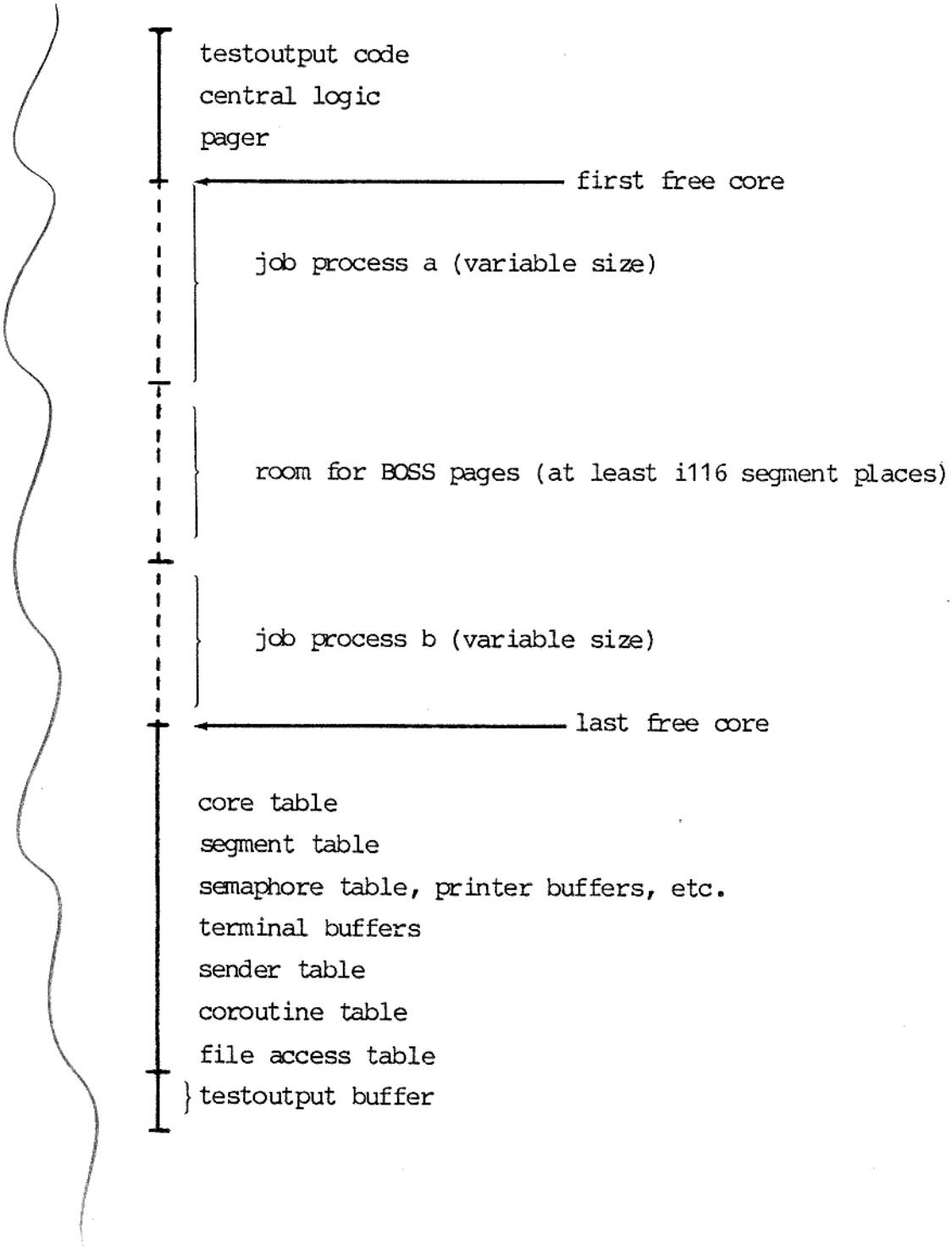
too few stations
too few temp entries
too few temp disc slices
too few convert operations
too few account operations
too few message bufferws
too few area processes
too few internal processes
too few suspend operations
too few temp drum entries
too few temp drum slices
too few remote readers
too few standard readers
too few protection keys

3. RUNTIME REQUIREMENTS 3.

This chapter is a survey of the organization of BOSS at runtime. The information is useful for understanding the options (section 1.3).

3.1 Primary Storage Utilisation 3.1

At runtime the BOSS-process is organized like this:



Most of BOSS' code and variables is placed in the so-called virtual store consisting of two backing store files 'drumcore' and 'disccore'. Pages of the virtual store are transferred to/from the primary storage as the need arises. If only one or no job is present in core store, the the space thus left free will be used for pages. A large job may force the minimum room for pages to move to 'last free' (or 'first free'), thus also preventing another job from using the core store.

3.2 Resource Requirements 3.2

The resources required by BOSS depend on options. The requirements of a special installation may be estimated by means of the following rules of thumb. Please be aware, that these formulas only give an approximate result, due to changes and extensions.

The resource demands primarily depend on the following options:

number of pseudo jobs	i45
number of terminals	i4
(plus main console and operator display)	
number of standard printers	i71
number of remote printers	i190
number of tape readers	e52+e53
number of card readers	i93+i200
number of tape stations	e25+e26
number of exchangeable disc drives	e23

Resources for each item and fixed resources:

	primary storage (halfwords)	area processes	message buffers
pseudo job	56	1	1
terminal	82+i3*	1	1
std printer	56+e40**	1	1
remote printer	56+e47***	1	1
reader	56	1	1
card reader	56	1	1
tape station	32	0	1
exchangeable disc	22	0	1
fixed resources:			
resident code	2450	-	-
test buffer	512	-	-
preoccupied area processes	-	12	-
preoccupied message buffers	-	-	8

* terminal buffer size

** standard printer buffer size

*** remote printer buffer size

Besides the demand on primary storage as computed from the table above, BOSS requires room for the following three tables:

1. file access table, which describes BOSS' access to any area process in the monitor.
   size = 1 halfword for each area process in monitor.

2. segmenttable, which describes the segments of BOSS' virtual store. size = 1 halfword for each segment in drumcore and disccore (usually 600 - 1000 halfwords).

3. coretable, describing 'free core', (dotted line in 3.1), which is organized as a number of 'segment places'.
   coretable size = 4 halfwords for each segment place
   = (BOSS' process size
   - primary storage from table
   - file access table
   - segmenttable) / 516

The maximum size of a job will then be:

maxjobarea = BOSS' process size
- primary storage from table
- file access table
- segment table
- coretable size
- 512 * i116 (minimum area for BOSS' segmentation).

Apart from the resources mentioned, the BOSS process demands at least 4 internal processes (sparse, 8 will be better), plus the paper tape reader, plus the devices specified in the options, page 1. The paper tape reader must have the name 'reader', a possible card reader must have the name 'cardreader'.

The monitor constants for input time out from typewriters should be carefully considered. If BOSS expects input from a typewriter, all output to the typewriter will pend until the user (or operator) completes the input line or the monitor completes the operation because of time out.

Especially, messages from the operator to a terminal must await completion of input from the terminal, and operator requests from BOSS must await completion of input from the operator.

Disc slices:

BOSS demands a disc claim corresponding to the total disc slices as computed in the scheme plus resources to fulfil all rest claims in the user catalog. Of the total disc slices, i108*4 are available for jobs (see option i108 on page 3). Remaining disc slices are available for jobs.

Drum slices: Analogous to disc slices, i112 replacing i108.

Catalog entries: Analogous to disc slices, i107 replacing i108.
The rest claims in the user catalog are defined as:
-first word in user catalog - present number of catalog entries occupied by users in the user catalog.
This accounts also for disc packs not mounted at present.

Message buffers:
Of the total number of message buffers needed, i109*4 are available for jobs. Of these, i109 are available for class 3 jobs, 2*109 for class 2 jobs, 3*109 for class 1 jobs, and 4*109 for class 0 jobs. Remaining message buffers are available for all jobs.

Area processes:
Of the total number of area processes needed, (i110-1)*4 are available for jobs. Of these, i110-1 are available for class 3 jobs, 2*(i110-1) for class 2 jobs, 3*(i110-1) for class 1 jobs, and 4*(i110-1) for class 0 jobs. Remaining area processes are available for all jobs.

3.3 Swopping, spooling 3.3

A job in core may be swopped out because it waits for operator or user action or because a high priority job wants the core. The swop-out consists of stopping the job process and copying the core area into a swop area on disc. A corresponding swop-in takes place when the core area is free or occupied by a low priority job. The swop area is created as a separate file at jobstart. The internal resource demands of the job are increased correspondingly.

I/O to some devices (terminals, tape reader, card reader, and printers) is spooled, which means that a backing store area (the spool area) is used to collect the information in case the producer of the information runs faster than the consumer. Most

of the spool areas in BOSS (namely those for primary output from jobs) are created as separate files at job start, and the resource demands of the job are adjusted accordingly. Spool areas for controlled i/o (readers and printers) are a fixed part of the virtual store (option i41, i52, i67). In case of off-line jobs, the resources needed to hold the job file cannot be attributed to a job initially, as the job specification is interpreted later. Thus, a set of resources is set aside for the purpose (option i162, i163).

3.4 Other Files Used at Run Time 3.4

Catalog:
BOSS scans the main catalog at job termination in order to clean away entries and areas left non-permanent by the job (can be avoided by an option in the user catalog).

A logout both 'temp' and 'login'-entries will be removed (this can not be avoided).

Usercat:
BOSS searches the user catalog at initialization, job start, and login. The user catalog is modified whenever a user changes his amount of permanent files. The user catalog is reserved all the time by BOSS.

Accountjob:
When BOSS is started up, the account job is enrolled automatically with the job file found in the file named accountjob. The account job is then suspended until the account file is to be processed (chapter 5).

Accountfile, accountf1:
BOSS collects account information in the file 'accountf1'. When this file is to be emptied, it is renamed to 'accountfile' (if possible) and the account job is activated. Meanwhile BOSS continues collecting in a new 'accountf1'. Further details in chapter 5.

Bosstest:
When BOSS starts up, it checks to see whether a magtape with the name bosstest exists. Then the tape is used for testoutput, starting at the beginning of the file on which the tape is positioned.

Otherwise BOSS tries to get access to a bs-file named bosstest. If it succeeds, testoutput is generated here. Otherwise the testoutput is just stored cyclically in the core store buffer.

The amount of testoutput is controlled by option e81 to e98 (see the options page 5).

4. BOSSOLD 4.

Bossold is the first of the source text files. This file contains a variety of programs which may be used in generating, translating, and updating BOSS.

The command

```
i bossold
```

generates the following programs (jobs):

```
bossbin
bossupdate
bosstrans
bossload
```

as temporary files (scope temp).

Bossold itself checks if the file descriptors are ok; if they are the program is terminated with the following message:

```
file descriptors left unchanged
```

In case no file descriptors exist for the files, file descriptors are created and the two files options and jobdescr are transferred to the bs device wanted.

4.1 Bossbin 4.1

Bossbin is a job which translates all the BOSS coroutines, the 'catupdate' program and the test output program.

4.2 Bossupdate 4.2

Bossupdate generates an updated version of all the BOSS files on a magnetic tape. The following commands may be used:

```
bossnew=set <mode> <tapename> 0 1
i bossupdate.
```

As a control of the files written on the magnetic tape the tape is check read and in the output it is possible to see if the generation of the tape has succeeded.

If 'bossupdate' cannot find a file descriptor named 'bossnew' the errormessage

- bossnew not set

will appear and the program will terminate.

4.3 Bosstrans 4.3

Bosstrans is much like 'bossbin' but only the BOSS coroutines are translated and stored on disc.

4.4 Bossload 4.4

Bossload loads all the files from the BOSS magnetic tape and stores the files as textfiles on the disc. The files get user scope. If temporary file descriptors exist with names equal to some of the files to be loaded these temporary entries are removed.

If bossold does not exist the following errormessage appears

- bossold not set

Bossold must exist because the files to be loaded are assumed to be loaded from a magnetic tape described by the file descriptor 'bossold'.

If 'bossold' exists on another document than a magnetic tape then the following errormessage will appear

- bossfiles not loaded

In case of an errormessage the program will terminate.

If a file 'bossnew' exists on a bs document, all the BOSS files will be loaded on this document, otherwise they will be loaded on the system disc.

5. ACCOUNTING 5.

While BOSS is running, essential information concerning the use of resources is collected as records in a file with the name 'accountf1'. This file is created and maintained by BOSS as a permanent file with the size given by the option i105 (options page 2). The file will be created on 'disc' and the entry base will be (-8388607, -8388607), which is called the 'account base'.

When BOSS starts up, 'accountf1' will be looked up, and if it does not exist, BOSS will create an empty file as described above. If the file exists, BOSS will scan it in order to find out where output to the file is to be resumed. Normally there will be a special stop-record, and output will then be resumed so that this record will be overwritten at the next writing. At the same time a new stop record will be inserted, except if current segment is full.

If, during the scan, BOSS finds a record with illegal contents, BOSS will assume that a crash happened, and resume writing from that point.

All records contain the system time of their production, and if the value of system time in the last record found during the scan is larger than the current value of system time, BOSS will stop with the message:

```
time inconsistence accountf1
***boss fault 5.
```

This is a valuable check on the initialization of system time during start up. In such cases, check if system time is correct. If not, autoload and reinitialize system time. Otherwise system time was not correct when BOSS was last running. In that case do as follows:

```
 *att ->s
->proc boss remove all boss run
  ready
```

```
   to boss
-> base 0 0 ; set catalog base = account base
-> rename accountf1.accountfile
-> base     ; reset catalog base (to system base)
-> bos      ; start boss
```

BOSS will then make a new 'accountf1' and enroll the accountjob for execution immediately after start up (cf. section 5.2).

You may also use another name than 'accountfile' e.g. (instead of "rename ..."):

```
-> rename accountf1.badaccount
-> scope user badaccount
```

and then analyse the file "manually". In that case you should not later rename 'badaccount' to 'accountfile' if BOSS has been started in the meantime. You might also use the standard accountjob (cf. section 5.2) to print the file and then cancel it, like this (instead of "rename ..."):

```
-> writeacc = algol message.no accountjob
-> o lp
-> writeacc in.accountf1
-> o c
-> scope temp accountf1
-> clear temp accountf1
```

Of course you may also make copies of 'accountf1', but apart from this, and obviously harmless variations of the examples given, it is not permitted to tamper with 'accountf1' or 'accountfile'. If you do, you may experience:

```
   ***boss fault 80
```

while BOSS is running, or:

```
   create accountf1 error
   *** boss fault 5
```

during start up. (Otherwise these errors will be due to errors in hardware).

5.1 Account Records 5.1

The accountfile contains fixed-length records (recordlength = 32 halfwords), blocked 16 records to a segment.

All records have the following format:

+0	project number (word)
+2 to +6	user name (9 characters) (not always terminated)
+8	user index etc. (word, see below)
+10	record kind (word)
+12 to +14	time (double word)
+16 to +30	parameters depending on record kind.

The contents of the field 'user index etc' are controlled by the option e71 (options page 5). If e71=0, (which is the default value) the word will always be 0 (providing a convenient terminator for the text in 'username'). If e71=1 the right half of the word will contain user index, while the contents of the left half of the word will depend on record kind (see below).

In the following list of existing record kinds, only the fields +8 and +16 to +30 are mentioned. If some of these fields are not mentioned, their contents are undefined. Please notice, that the addresses given here (and above) are relative to record start as they would be in a slangprogram. The standard account job (cf. 5.2) shows corresponding suitable field variable declarations and initializations for an ALGOL program.

Record kind = 1, jobfinis

Produced whenever a job finishes (this also applies to the small jobs which actually perform on-line converts).

+8 (hw) 0 or finis cause (see below)
+9 (hw) 0 or user index
+16 (word) net run time used (seconds)
+18 (hw) mounts used
+19 (hw) loads used
+20 (word) cpu-time used (unit:08192 secs)
+22 (hw) temp drum (slices demanded)
+23 (hw) temp disc (slices demanded)
+24 (hw) stations demanded
+25 (hw) size (unit: 512 hw)
+26 (word) device word (first 24 bits) (see below)
+28 (word) device word (last 24 bits) (see below)
+30 (hw) deliberate waiting (minutes)
+31 (hw) conversation (no. of lines input)

Finis cause is an integer code specifying how (or why) the job finished, the coding is:

0 (normal finis)
1 killed by user
2 time exceeded
3 replaced by jobfile: ...
4 job file exhausted
5 job file unreadable
6 killed by operator
7 mode unknown ...
8 output exceeded
9 not special station: ...
10 special station not ordred: ...
11 ring not allowed on tape: ...
12 tape reserved for other project: ...
13 stations exceeded at mount ...
14 suspends exceeded
15 login claims exceeded
16 tape used by other job: ...
17 accounts exceeded
18 convert file unreadable (see below)
19 max waiting time exceeded

20 bad fp
21 size too small for fp
22 no interrupt address
23 mount special after mount ...
24 corelock not allowed
25 corelock exceeded
26 load area in use ...
27 hard error on initial program or swop area ...
28 too few resources on ...
29 job name conflict
30 program does not exist
31 size too small
32 temp exceeded
33 replace file unreadable ...
34 replace file too long ...
35 temp exceeded job file
36 card deck exhausted
37 limited ...
38 option unknown ...
39 param at ...
40 syntax at ...
41 line too long after ...
42 device unknown ...
43 hard error on convert file ... (see below)
44 job creation impossible
45 file is no text file

If the job was an on-line convert, 100 will be added to the codes shown above, so that (e.g.):

100 means: on-line convert, normal finis
137 means: on-line convert, limited (disc slices).

The codes 18 and 43 will therefore never appear. (They are replaced by 118 and 143).

The device word (+26 and +28) is a double word with one bit corresponding to each device mentioned in the device list in options (options page 1, between e24: and e17=).

The most significant bit always equals 0, the next bit corresponds to job controlled reader, the next bit to job controlled printer, and so on backwards (i.e. from e17 and up) through the list, ending with special tape stations.

Using the standard options as an example, the device word is interpreted as follows:

0 shift 47 (always zero)
1 shift 46 job controlled reader (tapes ... ordred)
1 shift 45 job controlled printer (device printer)
1 shift 44 job controlled cards (device card)
1 shift 43 device punch
1 shift 42 device plotter
1 shift 41 device 14
1 shift 40 device 0
1 shift 39 device 10

Record kind=2, logout

Produced when a user logs out from a terminal. When the account job is being started (cf. 5.2) there is a brief period of time (some tens of milliseconds), during which the production of a logout record may be skipped.

+8 (hw) 0 or logout cause (see below)
+9 (hw) 0 or userindex
+16 (word) time logged in (minutes)
+18 (word) operations performed (see below)

logout cause is an integer with the meaning:

0 normal logout (requested by user)
1 hard error on terminal
2 operator remove
3 timeout on terminal

'Operations' is an estimate of the amount of work performed by BOSS during the session.

It is increased by 1 after:
- each command line and edit line read (also in autoline mode), but not for input lines to conversational jobs.
- each line printed in response to simple commands like 'display', 'list', 'verify', 'lookup', etc., but not for output from jobs.
- 'get', 'save', and 'transmit': one for each segment (768 characters) in the file.

Record kind = 3, print

Produced when a file has been printed.

+8 (hw) 0 or convert kind (see below)
+9 (hw) 0 or user index
+16 (word) number of lines printed
+18 (word) number of pages printed (see below)
+20 (word) paper type (see below)

'Convert kind' is an integer with the meaning:

0 normal convert, either on-line or from a job
1 primary output from a job
3 jobcontrolled printing

'Pages' is 1 + number of formfeeds in the file. 'Paper type' is the paper type ordered in the convert operation. For primary output 'paper type' will always be 0. For jobcontrolled output, 'paper type' is the last paper type used.

Record kind = 4, job (jobstart)

The option e70 (options page 5) controls the production of these records. If e70=0, which is the default value, they are not produced. If e70=1 the record will be produced at jobstart.

+8 (hw)	0	+17 (hw)	initial priority
+9 (hw)	0 or user index	+18 (word)	expected net runtime
+16 (hw)	0	+20 (word)	expected gross runtime

The record is produced when the job specification has been checked, and it has been checked that BOSS has resources to fulfil the claims of the job. This will normally be a few hundred milliseconds (typically 200-300 msec.) after the command starting the job was issued (or the paper tape or the card file containing the job was read). Notice that this is before reservation of resources, and before load-specifications (if present) are checked and executed.

The accountrecord is not produced for jobs which are killed or stopped because of errors before they reach the stage explained above. The account record is also not produced for the small jobs which carry out on-line converts.

In such cases a jobfinis accountrecord (kind = 1) may or may not be produced, depending on why the job finished: if errors occur before username, userindex, and project number have been checked (and approved) no jobfinis accountrecord will (or indeed can) be produced.

When the accountjob is being started (cf. 5.2), because the operator started it or because the accountfile is full, there is a short period of time (some tens of milliseconds), during which accountrecords of kind 4 cannot be produced (cf. accountrecords of kind 2).

Record kinds between 4 and 99: illegal.

Record kind = 99, stop
This record signals the end of the accountfile, as already mentioned. Please notice that all other fields than record kind are undefined.

Record kind >=100, private accountrecords
These records are produced, at the request of jobs, by means of the parent message 'account', which may be sent by means of the utility program 'account' or by other programs.

+8 (hw) 0
+9 (hw) 0 or user index
+16 (word) word m+14 from the account message.
+18 (word) word m+8 from the account message.
+20 (word) word m+10 from the account message.

The meaning of the information is defined by the user. Notice that record kind may also carry information.

If you use the utility program 'account', the connection between parameters and accountrecord will be: param1 = record kind, param2 = record +16, param3 = record+18, param4 = record+20.

5.2 Accountjob 5.2

The information collected in 'accountf1' may be used to supervise computer utilization, for resource planning, customer changing, etc. In order to take care of this, or at least to empty the file when it runs full, all installations must include an accountjob with these characteristics:

- The jobfile must be a permanent file with the name 'accountjob', it must be visible from the account base (-8388607, -8388607), and it must be present when BOSS is started.

- The project must be defined in the user catalog with a project base which includes the account base, and it must (at least) have permanent resources on the system disc to account for the accountjob and two accountfiles.

- The user must have 'job width'=1, 'no. of user indices'=1 and the user base must be the account base.

- The accountjob must include actions to cancel or ortherwise dispose of (e.g. by renaming) a file with the name 'accountfile'.

Besides this the accountjob and the accountproject may be handled as ordinary jobs and projects.

The accountjob is looked up and taken through the first stages of a normal jobstart each time BOSS starts up, and it is then stopped just after the point where the jobstart-record (cf. 5.1) is produced. Then it waits until the account file runs full or the operator types 'start account'. When this happens, BOSS will rename 'accountf1' to 'accountfile' and let the execution of the accountjob continue.

Meanwhile BOSS will create a new 'accountf1' and use this file to collect the accountrecords.

When the accountjob finishes, BOSS checks that 'accountfile' has been removed, and initializes the accountjob as during start-up.

If 'accountfile' still exists when the accountjob finishes, BOSS writes the message

```
   accountjob not ok
```

and enrolls the accountjob for execution again.

The standard accountjob supplied with BOSS is shown in subsection 5.3 as an example of an accountjob.

If you want to use it or modify it to suit your own needs, please notice:

- The job may run with a size as small as 13312 hw but inefficiently.

- The job will remove 'accountfile', even if the printing of the accountfile is unsuccessful.

- You may want to change the value of lines_on_a_page, slicelength_on_drum, and slicelength_on_disc.

If you make your own accounting system, you should try to minimize the runtime of the accountjob, and let an ordinary job carry out the main part of the work, e.g. as outlined in the following rough sketch:

The accounting project in the user catalog:

```
10 4711 0 10 3 -8388607 -8388604
 4 perm disc1 50 20 21
 5 perm disc1 0 0
11 accouser 4711 0 1 1 -8388607; userbase = account base
; notice, that 'boss private base' (-8388606, -8388606)
; is - and must be - left free.
11 otheruser 4711 0 1 2 -8388605
; notice, that this user has 2 userindices, one for
; 'bigjob' and one for editing etc.
```

The file 'accountjob' contains something like this:

```
job accouser 4711
mode list.yes
lookup accountfile accountf1 accofile
rename accountfile.accofile
scope project accofile
newjob bigjob
finis
```

The file 'bigjob' contains:

```
job otheruser 1 4711 time... etc.
...
... this is where the real work is done
...
finis
```

Finally: if you want to use the jobstart records (kind=4) to keep track of job turn-around-time, please remember that there is not a one-to-one correspondence between jobstart records and jobfinis records, especially not if you only scan the accountrecords in a single accountfile at a time.

5.3 The Standard accountjob 5.3

The following is a listing of the file 'taccount' which contains the standard accountjob.

```
; jma 1980.07.10          boss2, taccount          ...1...

accountjob=edit

i&
job account 51 size 16384 time 6 0 0 temp disc 800 8
head
mode list.yes
lookup accountfile accountf1

(
o accountlog              ; delete this line,
writeacc=set 40 1         ;    and this line,
writeacc=algol            ;    and this line,
writeacc in.accountfile   ;    and this line,
o c                       ;    and this line,
convert accountlog std 2  ;    and this line, if log is not wanted.
scope temp accountfile
clear temp accountfile
finis
)

begin
  message accountjob, versionid: 80 07 10, 9;
  boolean nl, sp;
  integer i, date, clock,
          lines_on_a_page, line, page,
          cause, devi1, devi2, type,
          slicelength_on_drum, slicelength_on_disc;
  integer array dummy(1:1);
  real t, k;
  real array fpparam(1:2);
  zone z(128*2, 2, stderror);
  boolean field userindex,
                finiscause, mounts, loads, tempdrum, tempdisc,
                stations, size, waitingtime, onlinelines,
                logoutcause,
                converttype,
                priority;
  integer field project, kind,
                netruntime, cputime, deviceword1, deviceword2,
                logintime, operations,
                lines, pages, paper,
                exp_netruntime, exp_grossruntime,
                param1, param2, param3;
  long field time;
  real array field username;

procedure newline(lines_needed);
  value lines_needed;
  integer lines_needed;
  begin
    if line + lines_needed > lines_on_a_page then
    begin
      page:= page + 1;
      write(out, <:<12>account file log:>,
      sp, 85, <:page:>, << ddddd>, page, nl, 2,
      <:date   time     user (index) project kind   (cause)  :>);
      if page = 1 then write(out, nl, 2,
        sp, 37, <:jobfinis        :>);
      write(out,
      <:net   cpu   size  drum  disc  load  stat mount  wait lines:>);
      if page = 1 then write(out, nl, 1,
        sp, 37, <:logout          min  oper:>, nl, 1,
        sp, 37, <:print         lines pages  paper:>, nl, 1,
        sp, 37, <:job             net gross   prio:>, nl, 1,
        sp, 37, <:(private)    param1       param2      param3:>);
      write(out, nl, 1);
      line:= if page = 1 then 10 else 4;
    end;
    write(out, nl, 1);
    line:= line + 1;
  end;
```

```
<* initialization *>
<* common fields                                          *>
project         :=  2; <* word          *>
username        :=  2; <* text, 9 chars *>
userindex       := 10; <* right hw      *>
kind            := 12; <* word          *>
time            := 16; <* doubleword    *>

<* fields in jobfinis-record              (kind = 1)    *>
finiscause      :=  9; <* left  hw      *>
netruntime      := 18; <* word          *>
mounts          := 19; <* left  hw      *>
loads           := 20; <* right hw      *>
cputime         := 22; <* word          *>
tempdrum        := 23; <* left  hw      *>
tempdisc        := 24; <* right hw      *>
stations        := 25; <* left  hw      *>
size            := 26; <* right hw      *>
deviceword1     := 28; <* word          *>
deviceword2     := 30; <* word          *>
waitingtime     := 31; <* left  hw      *>
onlinelines     := 32; <* right hw      *>

<* fields in logout-record                (kind = 2)    *>
logoutcause     :=  9; <* left  hw      *>
logintime       := 18; <* word          *>
operations      := 20; <* word          *>

<* fields in print-record                 (kind = 3)    *>
converttype     :=  9; <* left  hw      *>
lines           := 18; <* word          *>
pages           := 20; <* word          *>
paper           := 22; <* word          *>

<* fields in jobstart-record              (kind = 4)    *>
priority        := 18; <* right hw      *>
exp_netruntime  := 20; <* word          *>
exp_grossruntime:= 22; <* word          *>

<* fields in stop-record                  (kind = 99)   *>
<* only common fields *>

<* fields in private account records      (kind >= 100) *>
param1          := 18; <* word          *>
param2          := 20; <* word          *>
param3          := 22; <* word          *>

<* remaining initialization *>
nl:= false add 10;
sp:= false add 32;
lines_on_a_page:= 48;
line:= lines_on_a_page;
page:= 0;
slicelength_on_drum:= 1;  <* installation dependent *>
slicelength_on_disc:= 1;  <* installation dependent *>

<* first fp parameter in call is irrelevant *>
<* second parameter must be point and filename *>
if system(4, 2, fpparam) <> 8 shift 12 + 10
  then system(9, 2, <:<10>***param:>);
i:= 1;
open(z, 4, string fpparam(increase(i)), 0);
```

```
rep:
  inrec6(z, 32);
  if z.kind = 99 then goto stop;
  newline(if z.kind = 1 then 3 else 1);
  k:= z.time//10000;
  date:= systime(4, k, t);
  clock:= t;
  write(out, <<dddddd>, date,
    if clock >= 1 00 00 then << dd dd dd> else <<  z dd dd>,
    clock, sp, 1);
  outtext(out, 9, z.username, 1);
  write(out, << bd>, z.userindex extract 12,
    << dddddd>, z.project, sp, 1);
  if z.kind >= 1 and z.kind <= 4
  then write(out, case z.kind of (
    <: jobfinis:>, <: logout  :>, <: print   :>, <: job     :>))
  else write(out, << -ddddddd>, z.kind);
  write(out, sp, 1);

  if z.kind = 1 then
  begin
    comment jobfinis;
    cause:= z.finiscause extract 12;
    devi1:= z.deviceword1;
    devi2:= z.deviceword2;
    if cause = 100 then write(out, <:conv:>)
                   else write(out, <<bddd>, cause);
    write(out, << ddddd>,
      z.netruntime,
      z.cputime*0.8192,
      << dddddd>, (z.size extract 12)*512, << ddddd>,
      (z.tempdrum extract 12)*slicelength_on_drum,
      (z.tempdisc extract 12)*slicelength_on_disc,
      z.loads extract 12,
      z.stations extract 12,
      z.mounts extract 12,
      z.waitingtime extract 12,
      z.onlinelines extract 12);
    if devi1 <> 0 or devi2 <> 0 then
    begin
      write(out, nl, 1, sp, 37, <:deviceword1: :>);
      line:= line + 1;
      for i:= -23 step 1 until 0 do
      write(out, if devi1 shift i extract 1 = 1
                 then <:1:> else <:.:>,
                 sp, if i = -12 then 1 else 0);
      if devi1 shift (-22) extract 1 = 1
        then write(out, <: reader:>);
      if devi1 shift (-21) extract 1 = 1
        then write(out, <: printer:>);
      if devi1 shift (-20) extract 1 = 1
        then write(out, <: card:>);
      if devi1 shift (-19) extract 1 = 1
        then write(out, <: punch:>);
      if devi1 shift (-18) extract 1 = 1
        then write(out, <: plotter:>);
      if devi2 <> 0 then
      begin
        write(out, nl, 1, sp, 37, <:deviceword2: :>);
        line:= line + 1;
        for i:= -23 step 1 until 0 do
          write(out, if devi2 shift i extract 1 = 1
                     then <:1:> else <:.:>,
                     sp, if i = -12 then 1 else 0);
      end
    end
  end
  else
```

```
  if z.kind = 2 then
  begin
    comment logout;
    cause:= z.logoutcause extract 12;
    if cause = 1 then write(out, <:hard:>) else
    if cause = 2 then write(out, <:remo:>) else
    if cause = 3 then write(out, <:time:>)
                 else write(out, <<bddd>, cause);
    write(out, << ddddd>,
      z.logintime,
      z.operations);
  end
  else
  if z.kind = 3 then
  begin
    comment print;
    type:= z.converttype extract 12;
    if type = 1 then write(out, <:prim:>) else
    if type = 3 then write(out, <:jobc:>)
                else write(out, <<bddd>, type);
    write(out, << ddddd>,
      z.lines,
      z.pages, sp, 1,
      z.paper);
  end
  else
  if z.kind = 4 then
  begin
    comment jobstart;
    write(out, sp, 4, << ddddd>,
      z.exp_netruntime*0.8192,
      z.exp_grossruntime*0.8192, sp, 1,
      z.priority extract 12);
  end
  else
  if z.kind >= 100 then
  begin
    comment private account;
    write(out, sp, 4, <<    -ddddddd>,
      z.param1, sp, 1,
      z.param2,
      z.param3);
  end
  else
  begin
    comment illegal account kind;
    write(out, <: *** illegal account kind:>);
  end;

  goto rep;

stop:
  write(out, <:<12>end accountfile log:>);
end

finis

&, f

scope user accountjob
base
```

6. THE USER CATALOG IN BOSS 6.

General

At start up the operating system BOSS reserves the file named 'usercat' which must be visible from the account catalog base (see chapter 5). While BOSS is running, 'usercat' is used to identify users and determine their standard resources and possible maximum resources.

The file 'usercat' (the user catalog) is a hierarchical structure of subcatalogs of variable-length records. Each record describes a piece of information for a user or a project available in the user catalog. The sequence of records is as follows with the sequence shown in more and more detail:

```
first project      | project head
                   | project spec.
second project     | first user        | user head
      .            | second user       | user spec.
      .            |      .            |
last project
end record
```

The project head states the project number, the project catalog base and the project pool on disc. The user head states the user name, the user's catalog base and the number of user indices. Project specifications and user specifications may be absent, but if present they specify standard and maximum resources.

6.1 Resource Allocation to a Job 6.1

The resources available to a job are determined partially from the information found in the user catalog and partially from BOSS options. The actual resources for a job are determined by the following algorithm:

```
resources:= standard resources from options
modify resources according to project specifications
-     -       -       - user    -
-     -       -       - job     -
```

The final resources must be within the maximum resources and within the resources available to the time class of the job (cf. User's Manual, section 4.2)

6.2 The Structure of the User Catalog 6.2

The user catalog is sorted so that the projectnumbers appear in ascending order, and within the projects the user names appear in ascending order.

6.2.1 Index Segments 6.2.1

The first few segments of the user catalog contain an index table in which word no. n describes the contents of segment no. n. A negative number means that the segment is part of the index table, while a positive number is a project number, which indicates that the last information on the corresponding segment concerns this project number.

Unused words in the last part of the index table have the value 8388607 (the maximum positive integer).

Furthermore the first word in the index table is used to store the total number of entries claimed by users and projects in the catalog (represented as a negative number).

The end of the user catalog is signalled by a record of type 0 with project = 8388607.

6.2.2 Record Types in the Catalog

The two first halfwords of each record describe the type and the length of the record.

Type 0 and type 2 records are special records. Type 0 describes projects and type 2 users within the projects. All common information and resources follows a type 0 record and all information concerning a user follows a type 2 record. Type 0 and type 2 records are mandatory in the catalog, all other types are optional.

The following lists all the record types and for each type the meaning of each field (halfword or word) is described.

Many of the records correspond to options in the job specification. These records may indicate either a standard value (if the type is specified as an even integer) or a maximum value (if the type is incremented by one (odd value)). Such record types are marked with an asterisk (*).

type 0	project (input: type 1 and 10)
+0	0, 12
+2	project number
+4,6	project interval
+8,9	rest entries, rest slices on disc
+10,11	total entries, total slices on disc

type 2	user (input: type 2 and 11)
+0	2, 16
+2-8	user name (8 halfwords)
+10	user interval start
+12	standard interval length
+14	number of user indices

type 4	priority / late (input: type 7 (no longer used))
+0	4, 6
+2	standard value of priority
+4	minimum value of late

```
 type 6    private disc kits (input: type 4)
 +0        6, 16
 +2-8      device name (8 halfwords)
+10,11     rest entries, rest slices
+12,13     total entries, total slices
+14        slice length

*type 8    devicemask (input: 5, 6 device)
 +0        8*, 6
 +2        first word with device bits
 +4        second word with device bits

*type 10   accounts (input: 5, 6 accounts)
 +0        10*, 4
 +2        number of account buffers

*type 12   area processes (input: 5, 6 area)
 +0        12*, 4
 +2        area claim

*type 14   message buffer (input: 5,6 buf)
 +0        14*, 4
 +2        buffer claim

*type 16   convert operations (input: 5, 6 cbuf)
 +0        16*, 4
 +2        internal claim

*type 18   internal processes (input: 5, 6 internals)
 +0        18*, 4
 +2        internal claim

*type 20   keys (input: 5, 6 key)
+0         20*, 4
+2         number of protection keys

*type 22   mounts (input: 5, 6 mounts)
+0         22*, 4
+2         number of mounts
```

```
*type 24  output (input: 5, 6 output)
+0        24*, 4
+2        number of slices in primary output

*type 26  size (input: 5, 6 size)
+0        26*, 4
+2        size of process (in halfwords)

*type 28  stations (input: 5, 6 stations)
+0        28*, 4
+2        number of standard tape stations

*type 30  tapes (input: 5, 6 tapes)
+0        30*, 4
+2        number of job controlled paper tapes

*type 32  time (input: 5, 6 time)
+0        32*, 4
+2        net run time (in units of 0.8192 secs.)

 type 34  user with userpool (input: type 9)
+0        34, 10
+2,4      user max interval
+6,7      rest entries, rest slices on disc
+8,9      total entries, total slices on disc

 type 36  drum, key 1 (input: 5 temp drum)
+0        36, 4
+2        temp drum entries, slices

 type 38  disc, key 1 (input: 5 temp disc)
+0        38, 4
+2,3      temp disc entries, slices

 type 40  disc, key 3 (input: 5 perm disc)
+0        40, 4
+2,3      perm disc entries, slices
```

```
 type 42  output identification (input: type 3)
+0        42, variable length
+2        text, max. 300 characters

 type 44  drum, key 3 (input: type 8)
+0        44, 6
+2,3      rest entries, rest slices
+4,5      total entries, total slices

 type 46  drum, key 3 (input: 5 perm drum)
+0        46, 4
+2,3      perm drum entries, slices

 type 48  private disc kit (input: 5 perm <kitname>)
+0        48, 12
+2-8      device name (8 halfwords)
+10,11    perm <kitname> entries, slices

*type 50  max turn-around time (input: 5, 6 latest)
+0        50*, 4
+2        latest (in units of 0.8192 secs.)

 type 52  information for accountjob (input: type 12)
+0        52, 16
+2-14     project identification in textform (max. 20 chars).

 type 54  program (input: 5 program)
+0        54, 10
+2-8      name of program to be loaded (8 halfwords)

*type 56  available suspend buffers (input: 5, 6 suspends)
+0        56*, 4
+2        suspendings

*type 58  online (input: 5, 6 online)
+0        58*, 4
+2        conversational jobs allowed (1) / not allowed (0)
```

```
*type 60   corelock (input: 5, 6 corelock)
+0         60*, 4
+2         corelock time

*type 62   degree of information (input: 5, 6 minimal)
+0         62*, 4
+2         minimal yes (1) / no (0)

*type 64   priority (input: 5, 6 priority)
+0         64*, 4
+2         start priority factor

*type 66   deliberate waiting (input: 5, 6 wait)
+0         66*, 4
+2         maximum wait time swopped out

*type 68   catalog preservation (input: 5, 6 preserve)
+0         68*, 4
+2         preserve yes (1) / no (0)

*type 70   terminal user rights (input: 5, 6 privilege)
+0         70*, 4
+2         privilege bits.
```

6.3 How to Create and Update the User Catalog 6.3

This program, 'catupdate', is automatically translated whenever the coroutine files are translated (by means of the command 'i bossbin', cf. 1.2 and 4.1). At the same time a small binary file 'boptions' will be created. 'boptions' contains information concerning the installation (device numbers and backing storage devices), extracted from 'options'. This information is read and used by 'catupdate'.

6.3.1 Syntax for the Call of 'catupdate' 6.3.1

$$\left\{\text{<leftside>=}\right\}_{0}^{1}\ \text{catupdate}\ \left\{\text{cat.yes}\right\}_{0}^{1}\ \left\{\text{list.}\left\{\begin{matrix}\text{on}\\\text{off}\end{matrix}\right\}_{0}^{1}\right\}_{0}^{1}\ \left\{\text{<input file>}\right\}_{0}^{1}$$

```
<leftside> > ::= usercat
             /   <filename>

<input file> ::= <filename>
             /   <mode kind>
```

<filename> is a name with at most 11 characters.

```
<modekind> :: = trf
           /    tre
           /    tro
           /    crc
           /    crd
```

6.3.2 Explanation of Parameters to 'catupdate' 6.3.2

```
<leftside> ::= usercat
           /   <filename>
```

<leftside> is the resulting user catalog produced by 'catupdate'. BOSS assumes the user catalog to be on the file 'usercat' and reserves this file when it is running. Therefore the user catalog must either be updated while BOSS is not running or the updating may take place under BOSS producing a user catalog in a different file and this other file may later be moved or renamed to 'usercat'.

If 'catupdate' is called without <leftside> and without parameters the existing user catalog is listed in a crude form corresponding to the input format to 'catupdate'. If this kind of listing is produced when BOSS has been running, it will show the current values of rest claims for backing storage resources.

cat.yes
If cat.yes is present it is assumed that the catalog is made from scratch, otherwise it is assumed that the input specifies changes to the already existing user catalog in the file 'usercat'.

list.{on / off}

list.on has the effect that the input to 'catupdate' is listed as it is read.
list.off no input is listed, this is used as default.

<input file>
The input file from which the new catalog is created. (The input format is described in later sections). The input file can for instance be the final output from an earlier run of 'catupdate'.

6.3.3 Implementation of Subcatalogs 6.3.3

To be able to make input to 'catupdate' (i.e. to be able to make a user catalog), it is necessary to know how the hierarchy of subcatalogs are implemented.

Each subcatalog is represented by a pair of integers denoting an interval. The subcatalog represented by (l,h) is contained in the subcatalog (l1,h1) if and only if

$l1 \leq l \leq h \leq h1$.

The biggest subcatalog is (-8388607, 8388605) all other catalogs are subcatalogs of this catalog. For further explanation see the Monitor3 manual.

6.3.4 Input to Catupdate 6.3.4

The input in most cases consists of lines which specify a new record in the user catalog, a change of a record, or deletion of a record. Each line starts with a number specifying the record type. Spaces or tabulator separate the fields of the line.

Comments may be inserted in parenthesis between the fields, or at the end of the line after a semicolon.

The input lines must occur in a sequence corresponding to the records of the user catalog, thus:

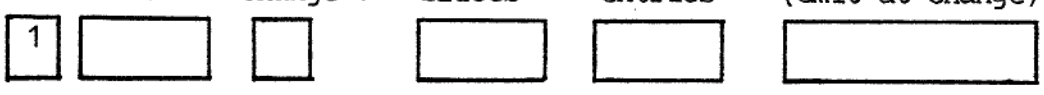
```
project head (type 1 or 10)
project specifications (types 3 to 8, 12 and up)

user head 1 (type 2 or 11)
user specifications (types 3 to 9, 12 and up)

user head 2 (type 2 or 11)
user specifications (.....)

...
end input (type -1)
```

However, all sorting of projects and users within projects are done by the program.

Below we show all possible line types. These lines may be composed into various input forms according to the needs of the staff. An example is shown later.

If you always make your user catalog up from scratch, it is recommended <u>not</u> to use line types 1 or 2, because it will cause you troubles when adding (or removing) projects or users. Instead you should use line types 10 and 11.

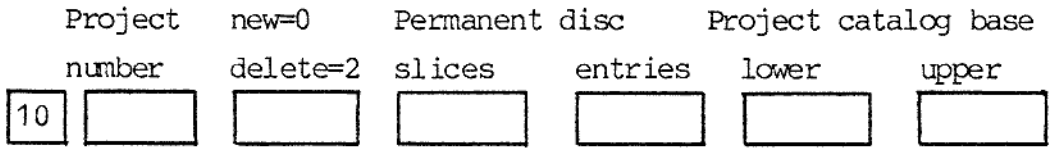
<u>Project head, simple</u>



This record may specify a new project and its permanent disc resources. The project number must be greater than 0 and less than 1000000 and the project catalog base is allocated automatically so that it does not overlap other projects (except maintenance projects). The width of the project catalog base is given in the last field. Normally, this should be 10 times the maximum number of users ever on this project. The width cannot be changed later.

The record may also specify a change of the permanent disc resources of a project (fields contain the changes, + or -), possibly followed by changes specified by other records.

Project head, abs base or deletion

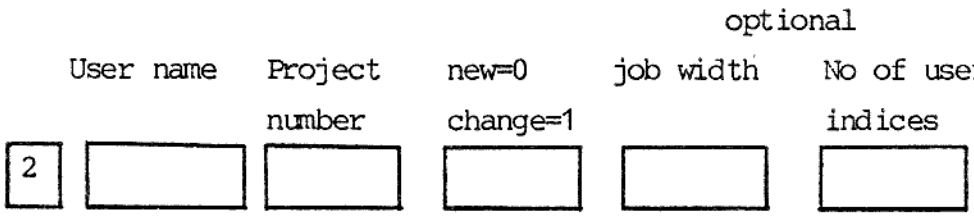


This record may specify a new project and its permanent disc resources. The project number must be greater than 0 and less than 1000000. The project catalog base must be supplied as a pair of numbers.

The record may also specify deletion of a project and all its users.

In this case the disc resources are not used. For safety reasons, the catalog base must be specified.

User head, simple

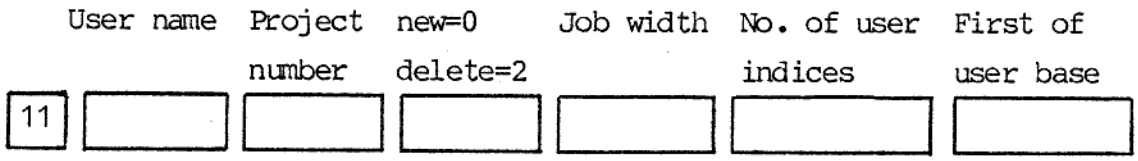


This record may specify a new user under the project. The user catalog base is allocated automatically so that it does not overlap other users of the project. The user catalog base will normally have a width of 10, allowing 10 user indices (10 jobs simultaneously from that user, independent of project). The standard base of the job will thus normally have a length of 1, but a larger width may be specified (job width).

User name must be one small letter followed by at most 8 small letters or digits. Notice that if the name is longer than these 9 characters, 'catupdate' truncates the name.

The record may also announce a change of the user specifications. The two optional fields may not occur in this case.

User head, abs base or deletion



This record may specify a new user under the project. The user catalog base is specified by its absolute first point. The last point will be: first of user base + job width * no of user indices - 1. The standard base of a job will be: (first of user base + job width * user index, same + job width - 1).

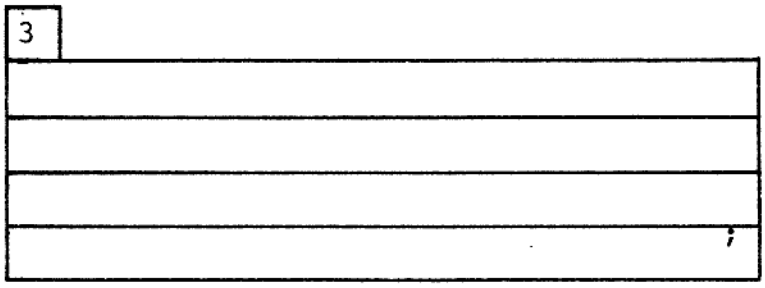
User name: see type 2 above.

Notice, that when a job terminates, BOSS cancels all temporary files within the standard base. Thus the abs base should be used with extreme care, especially if other users have their standard base in the base specified, or if it includes the private base of BOSS (-8388606, -8388606).

The record may also specify deletion of a user. For safety, the first point of the user base must be specified.

Output identification

The record specifies a textstring terminated with semicolon. Max. 300 characters. This text will appear on all print files. Leading spaces on a line are skipped. Leading underlines (ISO-value 95) may be used as significant spaces.

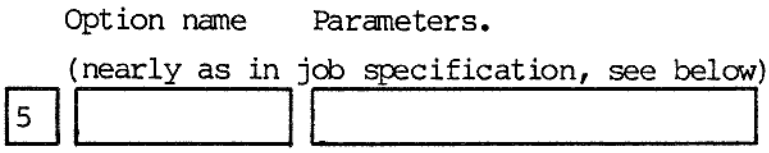


This record and the following types may appear after a project head (common for all users of the project) or after a user head (this user only).

Private kits

A job may only create areas on a kit (even temporary) if allowed by a private kit record. If this record is used after a user head, this user will not be able to use the project resources or change project files. <kit name> may not be 'disc' or 'drum'.

Standard value of option



'Device no' clears all previous device options. All device options in a set of project specifications or user specifications are combined and replace previous device information in the user catalog.

For the options 'output', 'temp', 'perm' the parameter must be specified as number of slices. Number of temp entries should be specified at most once. (i.e. either in 'temp drum' or 'temp disc', but not in both).

It is a good idea always to specify a standard value for permanent resources, in order to take care of the problem mentioned in User's manual for the perm-option.

Standard values for 'latest' and 'priority' can be specified.

Remember to change possible limit values of the options when a standard value is increased.

Limit value of option

	Option name (nearly as in job specification, see below)	Parameters.	Upper limit.
6			

Specifies the upper limit for options in the job specification. 'Device no' prevents the user from having any devices. He may be allowed to use specific devices by means of succeeding device options. All device options in a set of project specifications or user specifications are combined and replace previous device information in the user catalog. Limit value for 'output' must be in slices. Limit value for 'temp', 'perm', and 'program' cannot be specified.

Limit value for 'latest' and 'priority' can be specified. The limit value for 'latest' is a lower limit, while all other limits are upper limits.

Notice, that 'catupdate' does not check that the limit is consistent with the standard value or the actually available resources in the computer.

Drum

Permanent drum resources

	slices	entries
8		

User pool, no project right

Permanent resources on system disc

	slices	entries
9		

The user will have no right to change the project files, and will not be able to use the project resources.

Cannot occur in project specifications (i.e. between the project head and the first user head).

Accounting information

Further information for account job. Max. 20 characters. Terminate with semicolon. Leading spaces are skipped. Leading underlines (ISO-value 95) may be used as significant spaces.

12	;

End of user or project

-1

6.3.5 Example

6.3.5

Creation of a demonstration user catalog from scratch.

```
; BOSS demonstration user catalog
```

10	51	0	3	3	-8388607	-8388607

2	account	51	0	1	1

```
; a project No 51 is created on the account catalog base
; with the single user account (see chapter 6)
```

```
10  1  0  125  50  900  999
5   perm  disc  50  5
5   privilege  3
5   time  3  0
6   minimal  yes
6   online  yes
3   RC8000
      demonstration;
11  usera  1  0  1  20  900
11  userb  1  0  1  20  920
11  userc  1  0  1  20  940
; a new project 1 is created, it has 125 slices
; on the system disc and 50 entries in the
; catalog. The project catalog base is 900 to 999

; the project contains three users:
;     usera with standard base 900
;     userb  -     -     -  920
;     userc  -     -     -  940
; The explanation of the options can be found in
; the User's Manual.

10  2  0  50  10  1000  1020
5   time  5  0
6   online  yes
6   corelock  60
5   size  100000
6   preserve  yes
11  user d  2  0  1  10  1000
11  user e  2  0  1  10  1010

; a project 2 is created with two users
; who may specify that they
; want temporary files to be
; preserved after a job

-1
; end of input.
```

6.3.6 Errormessages from 'catupdate' 6.3.6

case out of range in testvalues
can be ignored.

delete not allowed
attempt to delete a userpool record which cannot be deleted

delete wrong interval
a deletion record tries to delete a project or user with a wrong project/user interval

device unknown
in the device option a name or a number of a device has been specified which is not present

different slicelength on same kit
a private kit is included with a slicelength which does not match an elsewhere given slicelength

existing
the record already exists, nothing has been done

illegal abs interval
the interval specified for a user is in conflict with the project interval

illegal claims
attempt to change claims to an illegal value

illegal devicename
disc and drum are reserved names and must not be used as a name of a private kit

illegal linetype
the linetype of an input record is greater than 12 or is less than -1

illegal projno
the projectnumber is out of range (linetype 1 or 10)

illegal parameter
a wrong parameter has been read from an option

illegal update identification
parameter 2 must be 0 or 1

illegal update information
should not appear

max. value not allowed
a maximum value is not allowed for this option

number of params
the number of parameters are out of range in a type 1 record

number out of range
one of the parameters in current record is an illegal number.

option unknown
an option specified in the input cannot be recognized, most likely because of a spelling error

out of sequence
illegal sequence of input record, most likely because a project- or user-record (type 1, 10, 2, or 11) was not accepted.

parameter kind
text where a number was excepted, or vice versa.

privilege illegal
the privilege class must be between 1 and 9

project unknown
the project number stated does not exist.

record unknown
attempt to delete an unknown record

std usercat index too small
the number of segments used as indexsegments is too small.
Catupdate must be changed

temp not allowed on specified device
only 'temp disc ...' and 'temp drum ...' is allowed

the catupdate program must be corrected and recompiled with a greater intervaltable size
selfexplanatory

too few parameters on line
a record has too few parameters, the linetype is neither 3 nor 12

too many parameters
an input record has too many parameters

too many priv kits
it is tried to reserve too many private diskkits
(option dependent)

updated
the record has been updated

user unknown
attempt to change or delete a non existing user

-1; end of catalog
the input has been transformed to usercatalog records, or if catupdate cannot find an inputfile this message will appear

6.4 Listing of User Catalog ('userout') 6.4

When file 16 of the BOSS text tape is loaded (by 'i tuserout' or the equivalent), an ALGOL program, 'userout', is produced on disc with user scope. When the program is called, it prints the current contents of usercat on current output. The program should be started on the accounting catalog base (see chapter 6).

6.5 Balancing Permanent Resources Against Actual Usage 6.5

As explained in chapter 3, BOSS during start-up computes the restclaims for users and projects having claims on the kits mounted. This computation is repeated for each kit which is later mounted by means of the 'kit'-command, so that BOSS at any time has full control over the resources: entries and slices on any kit. If BOSS detects an error in this accounting of resources one of the bossfaults 12, 17, 21, or 44 will result.

To avoid these errors you should stick to the following rules:

All resources occupied on backing storage should at any time be accounted for in the user catalog (do not let users make overdrafts).

If you have processes running in parallel with BOSS, their bases must not overlap those of a user logged in or a running job. Do not try to transfer files between such processes and BOSS-jobs or BOSS-users (and vice versa), i.e. if a file is created under s while BOSS is running, then do not change its size or scope under BOSS (and vice versa).

If you have projects or users with overlapping bases, preferably do not use them simultaneously. If you have to, then make sure to use 'preserve yes', but notice that 'preserve yes' is only a way to suppress a bossfault which ought to occur. And notice that 'preserve yes' does not work for a 'login', so do not log in under such a user or project.

For reasons of a technical nature you should also observe the following rules, especially for entries.

The commands finis, logout, save, scope or clear must not cause the total number of entries available for the job, the user, the project, or BOSS to pass the number 2047, i.e. either free entries must always be kept greater than 2047 or it must be kept lower than 2047.

It is quite simple to fulfil this rule:
Do not give the users many more entries, than they actually use, and adjust the size of the main catalog once in a while so that BOSS starts up with a reasonable number of entries, never more than about 1000.

Changing the size of the main catalog goes like this, when you have determined the new size wanted, let's say 197 segments:

```
  autoload the system
* att -> s
  clearcat
* att -> s
  maincat catalog 197
* att -> s
  oldcat
```

The new size of the catalog must be a few segments greater than the old size (each new segment holds 15 entries). The old size is determined by: lookup catalog.

The new size should not be divisible by 2 or 17, and the largest size allowed is 273.

7. HOW TO HANDLE A BOSS FAULT SITUATION 7.

If BOSS suddenly breaks down, because of an internal error in BOSS, a hardware fault, overbooked resources in the user catalog, or the like, it is important that the testoutput generated by BOSS, is saved for error documentation and correction.

If the testoutput is generated on disc (the area bosstest) the fastest way to get restarted is:

```
att s
remove all boss run
rename bosstest.testcopy
bosstest = set 200 disc
scope user bosstest
bos
```

Now the testoutput saved in the file 'testcopy' may be moved to a tape or analyzed and printed by the program 'last' (see 9.4.1)

If the error is to be reported to RC Computer then please send the testtape or the printer listing together with the error report.

8. BOSS FAULT ERROR CODES 8.

The alarm message 'boss fault' contains an error code giving a hint to the trouble. Error codes in the interval 1 to 9 originate from the coroutine file 'central', those in 10 to 19 from 'jobstart', and so on as specified below.

Central:

1 Program interrupt (break)
2 Access count < 0
3 Hard error, virtual store
4 No room in core for pages (i116)
5 Start-up alarm (always preceded by an explanation, see 2.2)
6 No message buffers
7 Access counter problems
8 - - -
9

Jobstart:

10 Trouble creating swop area
11 Trouble modify internal
12 Trouble term bs, all claims transferred to job
13 - - - , alarm create job
14 Trouble remove entry, load alarms
15 Trouble set cat base, catproc
16 Trouble reserve process, catproc
17 Common trouble, catproc, not done
18 Trouble, catproc, selective clear
19 Trouble, catproc, claim exceeded

Term1 and Term2:

20 BOSS closed normally by operator
21 Trouble after term bs, scope etc.
22 Trouble after prep access, save
23 Hard error termout, next job
24 Hard error termout, block size
25 Trouble term bs-adjust, logout
26 Trouble term bs-adjust, after transmit error
27 Trouble remove entry, at create output
28 Unknown error code, newjob
29

Jobstart:

30 Trouble remove primout
31 Trouble remove jobfile, kill
32 Trouble include user
33
34 Trouble request alarm
35 Trouble in catproc
36
37
38
39

Job:

40 Hard error primout, psjob i/o
41 FP not available, break and dump
42 Hard error catalog, clean catalog
43 Hard error primout, finis
44 Trouble after term bs, finis
45 Trouble after term bs-adjust, finis
46 Trouble after remove or adjust primout
47 Trouble after remove area process, finis
48 Illegal internal operation
49 Trouble adjust primout, finis

Mount:

50 Trouble include/exclude user
51 Illegal state
52 Trouble include user, check action
53 Trouble initialize process, name
54 Request trouble, psjob mount
55 Request trouble, remoter
56 Illegal station table change
57 - - - -
58 Device number trouble
59 Illegal station table change

Read:

60 Trouble reserving reader
61
62
63
64
65 Reader not device 0
66 Card reader error
67
68
69

Print:

70 Hard error in reading usercat
71
72 Primout does not exist or area does not exist
73 Date and time too long
74 Job controlled printing inconsistent
75
76
77
78
79

Procs:

80 Hard error, accountf1
81 Hard error usercat, output bsadjust
82 - - - , input bsadjust
83 - - - , init from usercat
84 Hard error catalog, unknown sender
85 - - - , - -
86 - - - , - -
87 Too many private kits
88 Trouble creating accountf1
89 Action from table outside limits, init from usercat

Banker:

90 Trouble priority queue, lock in core fit
91 Hard error swopout
92 Convert feature, not implemented
93 Trouble priority queue, release
94 Trouble res queue, simulate release
95 Too many free stations
96
97 Reserve reader, not found
98 Release reader, not found
99 Assign resources, reader not found

Term1 and term2:

100 Login attention during run
101 Login attention during logout
102 Trouble general print, integer size
103 - - - , buffer limit
104 - bs-adjust, snapshot
105 Page troubles, kitchange
106 Hard error usercat, kitchange
107
108
109

Mount:

110 Station table inconsistence
111 - - -
112 No station in searched state
113
114
115
116
117
118
119

Banker:

120
121 Job in core semaphore out of range
122
123
124
125
126
127
128
129

9. TESTOUTPUT 9.

The testoutput from BOSS consists of a sequence of variable-length, binary records stored in blocks of 512 halfwords. Output on magnetic tape is terminated by a tape mark (generated at close down). When output on backing store meets the end of the area, output continues in the first segment after the last type-14 record (see 9.1.14).

The BOSS text tape contains an external ALGOL procedure 'bossout' and a program 'last' which are used for analysis of the testoutput and output. These are described in section 9.4.

9.1 Record Format 9.1

Each record consists of a head and a variable length tail. The head is always 3 words, except for type 0, where the length is only one word.

head, word1: tail length shift 6 + type
head, word2: time
head, word3: 'third' meaning various things.

The tail length is given in halfwords. The time is given in units of 0.01 second from start-up of BOSS.

Type 0, next block

Shows that the remaining part of the block is unused. Tail length = 0. Notice that the head of this record is only one word.

9.1.1 Type 1, send 9.1.1

Produced when the following Central Logic entries are called: send and wait, send and wait fast, send and wait slow, stop and wait, wait answer.

Third = name table addr of receiver, tail length = 8.
tail: first 4 words of message. Tail makes no sense for the entries 'stop and wait' and 'wait answer'.

9.1.2 Type 2, lock 9.1.2

Produced when CL entries lock or lock chained are called.

Third = semaphore addr, tail length = 2
tail: semaphore value (before lock or lock chained).

9.1.3 Type 3, opch 9.1.3

Produced when open chained is called.

Third = semaphore addr, tail length = 8
tail: semaphore value (before open chained), first 3 words of operation (omitting the chain word).

9.1.4 Type 4, open 9.1.4

Produced when open is called.

Third = semaphore addr, tail length = 2
tail: semaphore value (before open).

9.1.5 Type 5, exit 9.1.5

Produced when Central Logic exits to a coroutine after all wait operations, after lock, lock chained, open chained, get pages, page jump, clear core. But not after open.

Third = coroutine ident (see 9.3), tail length = 18.
tail: core (x1 to x1+4), 5 absolute addresses of the pages, codepage ident (see 9.2) shift 12 + exit address relative to page 0.

Core (x1 to x1+4) contain the first 3 words of the answer after a wait operation, the first 3 words of the operation (including the chain word) after lock chained.

It is possible to generate a BOSS version where the 5 absolute addresses are replaced by virtual addresses (option e8).

9.1.6 Type 6, mess

Produced when a message is received from a job or a terminal. The record is produced when the coroutine is ready to receive the message, and this may be much later than the first detection of the message.

Third = coroutine ident, tail length = 8.
tail: sender process description address, first 3 words of message.

9.1.7 Type 7, answ

The record is produced the first time CL detects the answer.

Third = coroutine ident, tail length = 2.
tail: result of wait answer.

9.1.8 Type 8, jd-1

Generated whenever a jd-1 is executed.

Third = coroutine ident, tail length = 12.
tail: w0, w1, w2, w3, exception, instruction counter

On RC8000 'exception' is replaced with 'status' (see RC8000 Reference Manual, subsection 2.2.1).

9.1.9 Type 9, stop

Generated at close down, which always corresponds to an internal interrupt (intended or unintended).

Third = coroutine ident, tail length = 24.
tail contains the dumped registers:

	RC8000	RC4000
tail (1):	w0	-
2 :	w1	-
3 :	w2	-
4 :	w3	-
5 :	status	exception
6 :	ic	-
7 :	cause	-
8 :	addr	3440650
9 :	3440648	-
10 :	-(bossfault code)	-
11 :	page 0 address	-
12 :	ident, rel	-

'status' and 'exception' are explained in 'RC8000 Reference Manual', subsect. 2.2.1, where also 'cause' is explained (subsect. 7.1).

Bossfault code: see ch.8. Page 0 address and ident, rel are the same as in the last 'exit' (cf. 9.1.5 and 9.2).

Types greater than 9, private output may be produced by any coroutine. Present types are shown below.

Third = coroutine ident, tail length = any value.
tail: anything.

9.1.10 Type 10

Tail contains banker's description of a pseudo job.

Four consecutive records are produced to describe a psjob.

First record, tail length = 20

```
tail(1): chain, chain          ;priority or idle, resource or
                               ;deadly. jobs may be in no
                               ;chain.

                               ;REST CLAIMS:
tail(2): all, stations         ;all: pattern showing if
                               ;static resources should be
                               ;included: 4: converts
                               ;          2: account
                               ;          1: other
tail(3): entries, slices       ;temp disc
tail(4): std, remote           ;readers

                               ;STATIC RESOURCES:
tail(5): converts, accounts    ;
tail(6): buf, area             ;
tail(7): internal, suspends    ;
tail(8): entries, slices       ;temp drum
tail(9): deviceword 1          ;
tail (10): deviceword 2        ;
```

Second record, tail length = 12

```
                               ;RESERVED RESOURCES:
tail(1): all, stations         ;all: see first record
tail(2): entries, slices       ;temp disc
tail(3): std, remote           ;readers
```

```
                                    ;WANTED RESOURCES:
tail(4): all, stations              ;all: see first record
tail(5): entries, slices            ;temp disc
tail(6): std, remote                ;readers
```

Third record, tail length = 16

```
tail(1): first segm, top segm       ;coreplace
tail(2): 255- pr, priority          ;protection, priority factor
tail(3): gross run left             ;units of 0.8192 secs.
tail(4): net run left               ;-     - -      -
tail(5): arrival time               ;-     - -      -
tail(6): max. waiting               ;-     - -      -
tail(7): jobname(1)                 ;3 chars
tail(8): jobname(2)                 ;3 chars
```

Fourth record, tail length = 14

```
tail(1): jobname(3)                 ;3 chars
tail(2): jobname(4)                 ;3 chars
tail(3): virt. address              ;answer or dump name
tail(4): state, jobclass<1+corelock;state:
                                    ;0: skip (not in one of below
                                    ;   states
                                    ;1: resources wanted
                                    ;2: core wanted (after start
                                    ;   job operation)
                                    ;3: waiting (only account and
                                    ;   convert)
                                    ;4: killed in skip state
                                    ;5: in no queue (onlinejob
                                    ;   between runs)
tail(5): time to time out           ;units of 0.8192 secs
tail(6): time to wait swopped out   ;-     - -      -
tail(7): expected finishing time    ;-     - -      -
```

9.1.11 Type 11

9.1.11

Five consecutive records produced by the banker.

First record, tail length = 4

tail(1): psjob rel. for a-job (0 means no job in that place
tail(2): - - - b-job (0 - - - - - -

Next four records, all with tail length = 30

Currently free resources. All values given are one too high. One record is produced for each of the four job classes (0-3):

```
        ;DYNAMIC RESOURCES:
tail(1) :free stations
tail(2) :free temp disc (maincat) entries
tail(3) :free temp disc slices
tail(4) :free std readers
tail(5) :free remote readers

        ;STATIC RESOURCES:
tail(6) :free convert operations
tail(7) :free account operations
tail(8) :free mess bufs
tail(9) :free area procs
tail(10) :free intervals
tail(11) :free suspends
tail(12) :free temp drum entries (dummy)
tail(13) :free temp drum slices
tail(14) :free devices 1
tail(15) :free devices 2
```

9.1.12 Type 12

9.1.12

This type is used for various purposes, see below.

9.1.12.1 Jobstart 9.1.12.1

output of job description page, 23 records, tail length = 20
output of job file page, 23 records, tail length = 20

9.1.12.2 Job 9.1.12.2

On psjob finis page (page 46a) the part of job description page containing information about runtime left (d115+0 to d115+10) is output.

9.1.12.3 Procs Testoutput 9.1.12.3

2 or 3 records before and 1 or 3 records after all hostmessages. The format of the records is as shown below.

First record, tail length = 22 hw, call parameters:

```
tail(1): absref message area
tail(2): dev.name(1)
tail(3): dev.name(2)
tail(4): dev.name(3)
tail(5): dev.name(4)
tail(6): host no.                ;logical deviceno. of jobhost
tail(7): host id                 ;devicehost
tail(8): mode<12+kind
tail(9): timeout                 ;0: don't wait
                                 ;<128: wait up to <value> min
                                 ;>=128: wait till you get it
tail(10): catalog base (lower)
tail(11): catalog base (upper)
```

Second record, tail length = 16 hw, message:

```
tail(1): operation <12+function<1+addr.mode ;operation
         tail(1) = 4101: lookup process
         tail(1) = 4102: lookup device
         tail(1) = 4108: linkup remote
tail(2): first addr. of data
tail(3): last addr of data
tail(4): linkno <12 + hostno.        ;logical dev.no. in jobhost
tail(5): hostid                      ;devicehost
tail(6): dh. homereg<12+dh.netid.
tail(7): (irrelevant)
tail(8): (irrelevant)
```

Third record, tail length = 22 hw, data:

```
(third record is not always present)
tail(1) : mode <12 + kind
tail(2) : timeout <12 + buffers
tail(3) : buffersize                 ;halfwords
tail(4) : devicename(1)
tail(5) : devicename(2)
tail(6) : devicename(3)
tail(7) : devicename(4)
tail(8) : (irrelevant)
tail(9) : (irrelevant)
tail(10): (irrelevant)
tail(11): (irrelevant)
```

Fourth record, 16 bytes, answer: (only produced if monitor result = 1)

```
tail(1): device status <16 + link descriptor <12 + function result
        ;device status: bit 0: device unknown
                        -   1: device closed
                        -   2:
                        -   3:
                        -   4:
                        -   5: device driver not loaded
                        -   6: device reserved by another process
                        -   7: reservation rejected from AP
                               (only GAC interface)
```

```
;link descriptor:
          (value)  0: no link (subprocess) present
                   1: remote link (subprocess) present
                   2: local link (subprocess) present

;function result:
          (value) -1: sender stopped
                   0: function executed
                   1: device troubles (see device status)
                   2: device reserved by other host
                   3: no resources at jobhost
                   4: no resources at devicehost
                   5: timeout
                   6: device requested with higher
                      priority
                   7: some link was present (see link
                      descriptor)
                   8: device host unknown
```

tail(2): number of bytes in 'data'
tail(3): number of chars in 'data'
tail(4): dh. linkno <12 + hostno.
tail(5): dh. hostid
tail(6): dh. homereg <12 + dh. net-id
tail(7): [not used]
tail(8): [not used]

Fifth record, 22 bytes, data: (only produced if monitor result = 1 and local link is not present)

```
tail(1) : kind
tail(2) : max. buffers
tail(3) : max. buffersize [chars]
tail(4) : name(1)
tail(5) : name(2)
tail(6) : name(3)
tail(7) : name(4)
tail(8) : jh. linkno.                ;tail(8) - tail(11) are
tail(9) : jh. hostid                 ; only relevant for
tail(10): jh. homereg <12 + jh.netid ;'linkup remote'
tail(11): process description address (subprocess)
```

sixth record, tail length = 4 or 10 hw, return parameters:
tail(1) : name table address
tail(2) : logical device no.
tail(3) : device kind
tail(4) : mode
tail(5) : max. buffersize [halfwords]

If 'function' (cf. second record) was 'lookup process' tail length will be 4 and tail(1) = tail(2) = 0.

Otherwise the length will depend on the result. If everything was ok (no link was present before the call, and the link was created as requested), the sixth record will be as shown above.

If something went wrong, tail length will be 4, tail(2) = 0 and tail(1) will be 0 or 'process description address' (if the device was linked before).

9.1.13 Type 13 9.1.13

Produced during start-up when a coroutine file is loaded.

Tail contains:
name of coroutine file (4 words), first of coroutine table, first of semaphore table, first of sender table, checksum, double sum, version date, version number.

The record corresponding to 'bos' contains BOSS start-up time instead of first of coroutine table, first of semaphore table and 'load address for coroutine files' instead of 'first of sender table'. The last two words contain first and last address of the BOSS process.

9.1.14 Type 14

9.1.14

Produced during start-up after the first loader pass.

Third=2*external number of first external in the record.
Tail length=100.
Tail contains the corresponding part of the external table.

9.1.15 Type 15

9.1.15

Produced when command or jobstart has read a line. Tail contains the line.

9.1.16 Type 16

9.1.16

Testoutput from bs-adjust. One record for each bs-device.

Tail contains:
entry claim (key 0), slice claim (key 0)
...
entry claim (key 3), slice claim (key 3).

9.1.17 Type 17

9.1.17

Produced when a catalog entry is removed (i.e. after jobtermination and after logout) and when a diskit is mounted.

Tail contains the catalog entry.

9.1.18 Type 18

9.1.18

Produced each time the mounter is activated.

Tail contains an entry of the station table.
(A closer description of the station table can be found on page 3 in the 'mounter').

9.1.19 Type 19 9.1.19

Produced each time the mounter is activated.

Tail contains an entry of the tape table.
(A closer description of the tape table can be found on page 3 in the 'mounter').

9.1.20 Type 20 9.1.20

Generated when Central Logic entries prepare_access or terminate_access are called.

Tail contains:
nametable address of area process,
1 or -1 (depending on prepare_access or terminate_access), access count, interval of entry, name of entry.

9.1.21 Type 21 9.1.21

Area process description (see the Monitor 3 manual).

9.1.22 Type 22 9.1.22

Produced immediately before the stoprecord (9.1.9) at close down.

Tail contains i205 (options page 6, bottom) hw of the code ending with ic-2 (the instruction causing the interrupt). i205 is adjusted so that 20<= i205<=120.

9.2 Codepage Identifications

The codepage identifications used in testoutput records type 5 (see 9.1.5) and type 9 (see 9.1.9) are shown below.

central

9 : pager

tjobstart

11: codepage 1 (job birth, online convert)

12: codepage 2 (load commands, set claim)

13: codepage 3 (command reading)

14: codepage 4 (create job, request alarm)

15: codepage 5 (load, rb-comm)

tterm 1

20: command input

21: commandio bulk file

22: command print

23: snapshot, autoline

tjob

40: psjob i/o

41: psjob aux

42: psjob break

43: clean catalog

44: psjob finis

tmount

50: psmount

52: remoter

54: rewinder

56: watchdog

58: commandio, name, label

tread

60: tape reader

61: card reader

62: start card

63: pstape

64: pscard

65: psload

tprinter

70: paper description

71: central page

72: triangle page

73: main loop

tprocs

80: account

81: bsadjust

82: init from usercat

83: unknown sender

84: various procedures for remote devices

tbanker

90: main banker

91: core allocation

92: resource allocation

tterm 2

100: kill

101: go, run, job, newjob

102: rename, clear, scope

103: login, get, save

104: transmit

105: display

106: kit changer

107: term out

108: request display

9.3 Coroutine Identifications 9.3

The coroutines are identified as follows in the testoutput. We also show the coroutine file which generates the coroutine.

0	output during start-up	(central)
1	timer	(banker)
2	banker	(banker)
3	unknown sender	(procs)
7	watchdog	(mount)
8	request display	(term2)
9	pager	(central)
10	operator display	(term2)
20*	kit changers	(term2)
40*	printers	(printer)
60*	rewinders	(mount)
80*	remoters	(mount)
100*	commandios	(term1)
200*	psjobs	(jobstart)
300*	termouts	(term2)
400*	card readers	(reader)
450*	readers	(reader)

* as many as there are incarnations, numbered consecutively.

9.4 Selected Testoutput

9.4

Testoutput is always produced during the initialization. Testoutput is not produced during the run if BOSS has been translated with e9=0. e9 is an option, which is defined in 'options' on page 5, together with other options which may be used to select testoutput independently for each coroutine type.

The operator may change the selection of testoutput dynamically by means of the 'test' command (see Operator's Manual).

The value of e9 is regarded by the system as 3 bits. The bit with value 1 means output from Central Logic and jd-1 (type 1 to 9). the bit with value 2 selects type 10 and 15 output, the bit value 4 selects type 11 output. Type 13 and 14 output is always produced.

When e9<>0 but no testoutput medium is available, the records are still produced in the testoutput buffer.

9.5 Analysis of Testoutput 9.5

When file 17 of the BOSS text tape is loaded (by 'i testout' or the equivalent), an external ALGOL procedure, bossout, and an ALGOL program, last, are produced. They are generated on disc with user scope.

9.5.1 Last 9.5.1

The program, last, prints selected parts of a testoutput file in readable form on current output. The call is:

last <doc>.<file> $\left\{\text{.<blocks>}\right\}_{0}^{1}$ $\left\{\text{<s><first>.<last>}\right\}_{0}^{1}$ $\left\{\text{<s><anything>}\right\}_{0}^{1}$

<doc>	is the name of the magtape or the backing store file holding the testoutput.
<file>	is the file numer or 'bs' meaning backing store file.
<blocks>	is the number of testoutput blocks to be printed counted from the end of the testoutput, as the latest blocks are usually most interesting after a break down. If <blocks> is omitted, all blocks are printed.
<first >.<last>	selects an interval of coroutine identifications for printing.
<anything>	Anything obeying the syntax for FP-parameters. The rest of the line is just copied to the testoutput and can be used for identification purposes

The identification records (type 13) are always printed in order to verify the BOSS version used.

<u>Example</u>: The normal call will be: last bosstest.bs.
The following call prints only testoutput from commandios in the last 10 blocks.

last bosstest.bs.10 100.150 maintenance manual testoutput

The output may look like the following illustration:

```
bossout 15 08 74     25 08  file: bosstest.bs.10 100.101 maintenance manual testoutput
ident: rialtoindk        0                                                               } identifications
                                                                                           of installation
                                                    checksums         version date    start and top
                                                                      and number      address of boss

start-up: 15 08 74     22 39
ident: bos             124525 -447367       0 2139790  417903   140874      56      45426   196606 \
ident: bterm2          195918  194580  195230 4458885-5419715   130874     109                     |
ident: bterm1          195758  194396  195230 2710605-4621752   140874      75                     |
ident: bjobstart       195502  194204  195170 3940254 6681084   130874      87                     |
ident: bjob            195502  194132  195070 3899212 4574612   130874      90                     > type = 13
ident: bmount          195502  194132  195070 2842785 5383009   310774     115                     |
ident: bread           195358  193958  195070 2031369 7061990   100774      31                     |
ident: bprinter        195342  193914  195070 2740198-3680670   310774      34                     |
ident: bprocs          195294  192582  195070 1961589-6920979   140874      58                     |
ident: bbanker         195278  192552  195060 2729485-2143833    60874      55                     /

lock    5990  194366
mess    5990     101    10568       0       0       0
exit    5990     101        0       0   33488   65786   59642  195184       0       0   20  206
lock    5990  194414
exit    5990     101        0  194414 3338456   65786   59642  195184  195184       0   20  216
send    5990    8698    20480  194660  194660 4079114

answ    6093     101        1
exit    6093     101        0       2       3   65786   59642       0  195184       0   20  278
send    6093    8698    12288  194660  194738   46178

answ    7705     101        1
exit    7705     101        0      14      20   65786   59642       0  195184       0   20  308
  15    7705     101    15 tpf=edit options

exit    7705     101      101       2     645   65786   59642       0   47866       0   20 1000
send    7705    8698    12288  194660  194738   46178

answ    8055     101        1
exit    8055     101        0       6       7   65786   59642       0       0       0   20  308
  15    8055     101    devi 1

exit    8058     101      101       2     645   51450   59642       0       0   70394   14   92
exit    8058     101      101       2     645   65786   59642       0       0   70394   20  648
send    8058    8698    12288  194660  194738   46178

mess    9321     100     9780       0       0       0
exit    9321     100        0       0   32768   65786   53498  195174       0       0   20  206
lock    9321  194408
exit    9321     100        0  194408 3305688   65786   53498  195174  195174       0   20  216
send    9321    8662    12288  194580  194658   46178

answ    9372     101        1
exit    9372     101  2097152       0       0   65786   59642       0       0       0   20  308
send    9372    8698    12288  194660  194738   46178

answ    9443     100        1
exit    9443     100        0       4       6   65786   53498       0  195174       0   20  308
  15    9443     100    close

exit    9447     100      100       2     645   55546   53498       0  195174       0  100  763
stop    9447     100        0       0     763   53498   55546       0   56310     -20  100
end     32

type    time    third                tail of varying length. type 15 is printed
                100=main console     in text form, other tails in decimal form.
                101=first terminal
```

9.5.2 Bossout

This procedure assumes that the program has been called in this way (see 9.5.1):

<program> <doc>.<file> $\left\{.\text{<blocks>}\right\}_{0}^{1}$ <possible further parameters>

The procedure is called like this:

bossout(type, time, coruno, third, record, move, print)

It scans the testoutput in the file specified and in the last <blocks> blocks. If <blocks> is omitted, the whole file is scanned.

Each time it gets a record, it assigns the type, time, coroutine identification, and 'third' to the first four parameters (integers), and assigns the length of the record (in bytes) to record(0). Then it interrogates the parameter 'move' (boolean, Jensen's device) to determine if the record should be moved into the integer array 'record' from index 1 and on (moves when true). Finally the last parameter, 'print' (boolean, Jensen device), is interrogated. When the parameter is true, the record is printed. (The stop record (type 9) and the identification records (type 13) are always printed).

Example: In order to print all events from time 300 (seconds) to 400 after start-up of BOSS, proceed as follows:

```
pr = algol
begin integer k,t,c,th; integer array record(0:0);
bossout (k,t,c,th,record,false,
        300 00 <= t and t <=400 00);
end
pr bosstest1.bs
```

Example: If you want to examine the records yourself (e.g. making statistics on free resources), then make use of the 'move' parameter:

```
pr = algol
begin integer k,t,c,th;
      integer array record (0:256);
      boolean procedure statistic;
         begin integer field i;
               if k=11 and record (0) > 4 then
                  begin
                  for i:=2 step 2 until record (0) do
                    ... statistics ... on record.i
                  end;
               if k=9 then ... terminate statistics ...
         statistic:=false;
         end;
      bossout (k,t,c,th,record,true,statistic);
end;
pr bosstape.3
```

RETURN LETTER

Title: BOSS2 INSTALLATION AND MAINTENANCE RCSL No.: 31-D628

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

REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark