

Title:

RC3600 CATALOG SYSTEM
SYSTEM PROGRAMMER'S GUIDE

 **REGNECENTRALEN**

RC SYSTEM LIBRARY: FALKONERALLE 1 DK-2000 COPENHAGEN F

RCSL No: 43-GL7915

Edition: 78.09.05

Author: J.C.Olsen

Keywords:

Catalog system, file system, area process, subcatalog.

Abstract: This manual describes how to use the RC3600 Catalog System from assembler programs. Also the organisation of the disc(s) is described. The user must be familiar with the MUS system.

This manual replaces RCSL: 44RT1278

Copyright A/S Regnecentralen, 1978
Printed by A/S Regnecentralen, Copenhagen

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

1.	INTRODUCTION.....	1
2.	THE CATALOG SYSTEM.....	3
3.	DISC FORMAT.....	5
	3.1 Unit Format.....	5
	3.2 File Format.....	6
	3.3 Index Block Format.....	8
	3.4 Catalog Format.....	8
	3.5 Sub Catalog Format.....	9
	3.6 Catalog Entry Format.....	11
	3.7 The 'SYSSC' File.....	12
4.	CATALOG OPERATIONS.....	14
	4.1 Parameter Description.....	14
	4.2 Main Catalogs.....	15
	4.3 Sub Catalogs.....	15
	4.4 Set Entry.....	16
	4.5 Create Entry.....	17
	4.6 Look-up Entry.....	18
	4.7 Change Entry.....	19
	4.8 Remove Entry.....	20
	4.9 Create Area Process.....	21
	4.10 Remove Area Process.....	22
	4.11 Init Catalog.....	23
	4.12 Created Catalog Process.....	23
	4.13 Remove Catalog Process.....	25
	4.14 Get Unit Description.....	26
5.	AREA PROCESSES.....	27
	5.1 Control.....	27
	5.2 Transput.....	29
	5.3 Status Bits.....	32
6.	FORMAT OF PROCESS DESCRIPTIONS.....	33
	6.1 Area Process Description.....	33
	6.2 Catalog Process Description.....	34

7.	OPTION PROGRAMS.....	35
	7.1 Producing 'CATW'	35
	7.2 Init New Catalog	37
	Appendix A - Catalog System Survey	39

1. INTRODUCTION

1.

The RC3600 Catalog System makes it possible to divide a disc drive into smaller independent units, called files. These files are identified by name. The descriptions of the files, file descriptions, are kept in a catalog (which itself is a file) stored on the disc. A file description contains information about the file such as name, length, starting position. The catalog consists of a number of catalog entries. Every file description is kept in a catalog entry.

The RC3600 Catalog System deals with units. A unit may be a disc drive, a part of a disc drive or include several disc drives. Each unit has its own main catalog, describing the files contained on that unit. Files on different units may therefore have the same names.

A file may be a new catalog (sub catalog) for a number of files. To access a file an area process must be created on that file. The area process for a file is initialized according to the catalog entry holding the file description. Input and output operations on a file is then carried out by sending messages to the area process.

Operations involving reading or writing the catalog file is done by sending messages to a process named 'CAT'.

To access files in a sub catalog, a catalog process must first be created for that sub catalog. Creation (and removal) of catalog processes is done by the 'CAT' process. All operation on the sub catalog and the files within it, are carried out by sending messages to the catalog process for that sub catalog.

The information of the units (main catalogs) is kept in a separate process named 'CATW'. These informations may have to be changed according to the application.

The necessary number of area processes (used to access files and sub catalogs) must be loaded seperately. Option programs containing 2, 3, 4 or 8 area processes are available and may be used in any combination.

The RC3600 Catalog System is able to handle all existing disc types which belong to the RC3600 hardware system.

2. THE CATALOG SYSTEM

2.

Before a new disc drive can be used by the RC3600 Catalog system, it must be initialized by a special utility program (see chap. 7.).

When a disc drive is to be used by the catalog system it must be initialized (see 4.11.). The catalog system will then read a specific sector on the disc drive to get the information necessary for using the unit(s) on that drive.

When a disc drive has been initialized the catalog system is able to perform the following operations concerning that drive:

- 1) set, create, lookup, change and remove catalog entries.
- 2) create and remove area processes
- 3) create and remove catalog processes
- 4) read and write on the files.

A file consists of a file description contained in a catalog entry, and a number of slices (≥ 0). A slice equals a fixed number of sectors. If the number of slices assigned to a file is greater than or equal to one, then the first sector in the first slice is an indexblock for the file. The indexblock contains information about where to find the slices assigned for the file. The first slice (which contains the indexblock) is pointed out by the catalog entry containing the file description for the file. The length of a file is given in number of sectors. The indexblock is not counted as a part of the file (i.e. filelength = 1, means one data sector). All operations on the file which involves changing of the filelength ensures that the number of slices necessary for the filelength is assigned to the file.

The total number of sectors occupied by a file (number of sectors pr. slice * no of slices assigned for the file) is called the reserved length.

When writing on a file, the reserved length may be exceeded. If so, the catalog system may extend the file with an extra slice.

Catalog processes are able to perform all operations except for creating (removing) new catalog processes.

The unit description block (also called kit description) contains information necessary for the catalog system to handle that unit. The format is:

Unit description block = sector 8

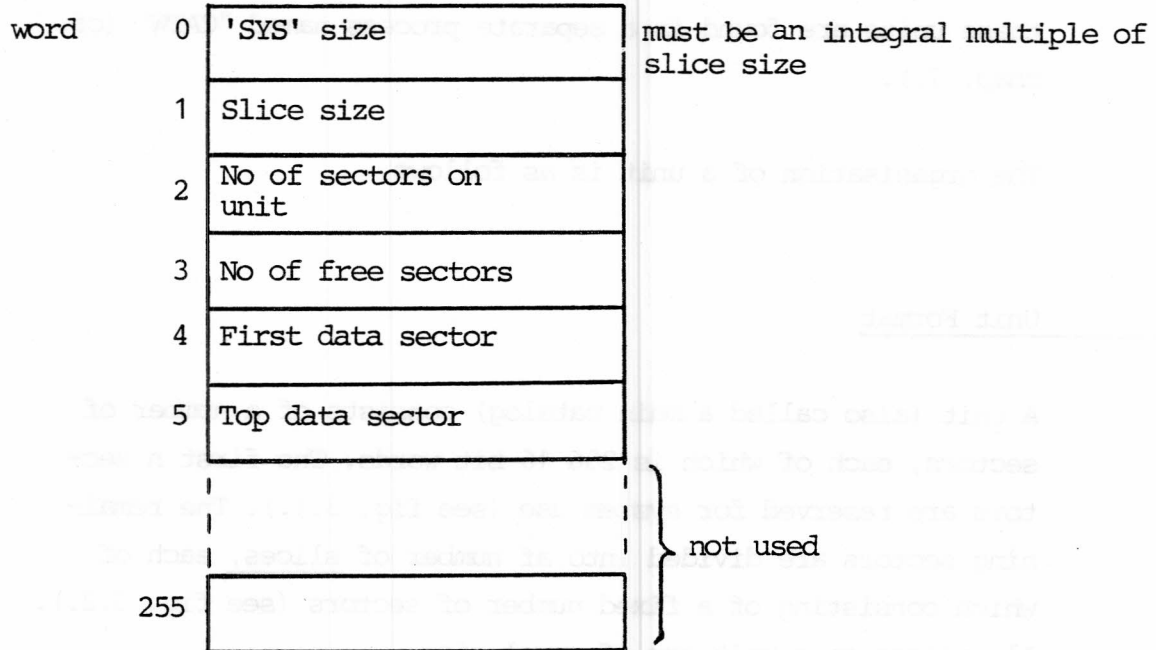


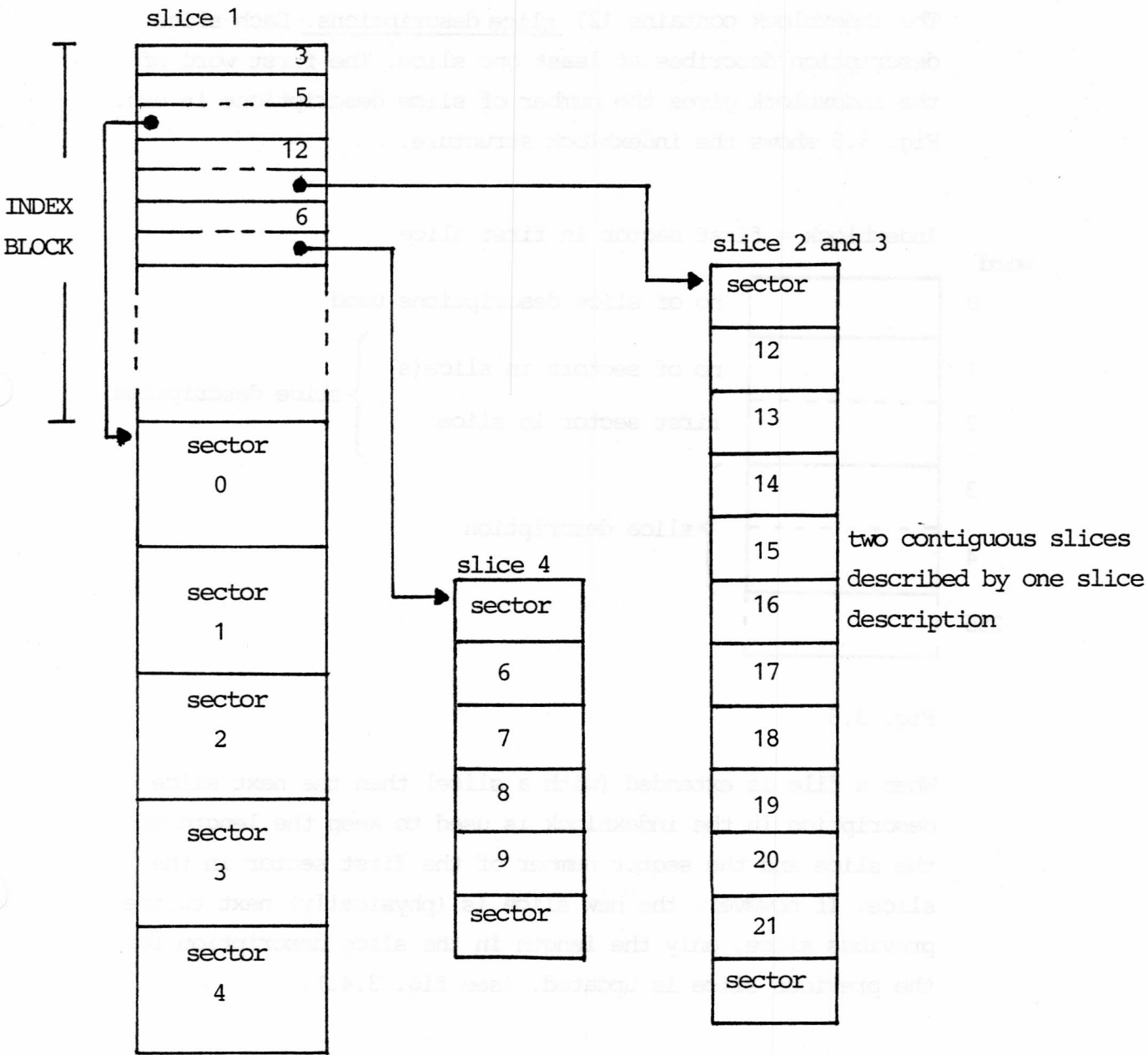
Fig. 3.3

The map block (see fig. 3.1.) contains a mapping of free/used slices on the unit. Each bit equals one slice such as bit 0, word 0 = slice 1. If a bit is zero the corresponding slice is used otherwise the slice is free.

3.2. File Format

A file consists of a number of slices which are linked together by means of an indexblock. The indexblock is always written in the first sector of the first slice, except for the catalog files 'SYS' and 'MAP' (see fig. 3.1.). The file structure is shown in fig. 3.4.

Fig. 3.4 File structure



A file is logically contiguous by means of the indexblock. The indexblock contains 127 slice descriptions. Each slice description describes at least one slice. The first word of the indexblock gives the number of slice descriptions in use. Fig. 3.5 shows the indexblock structure.

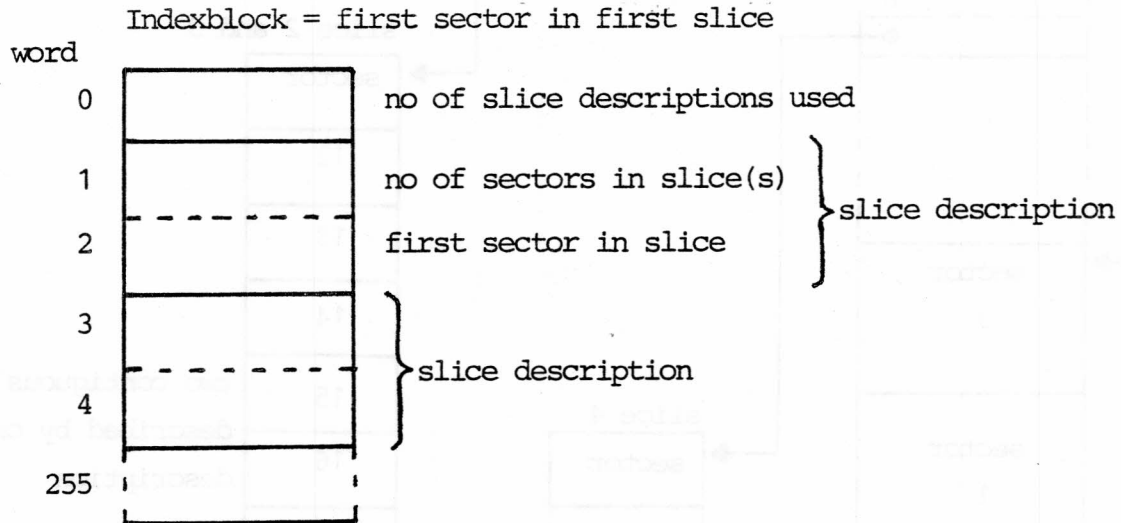


Fig. 3.5

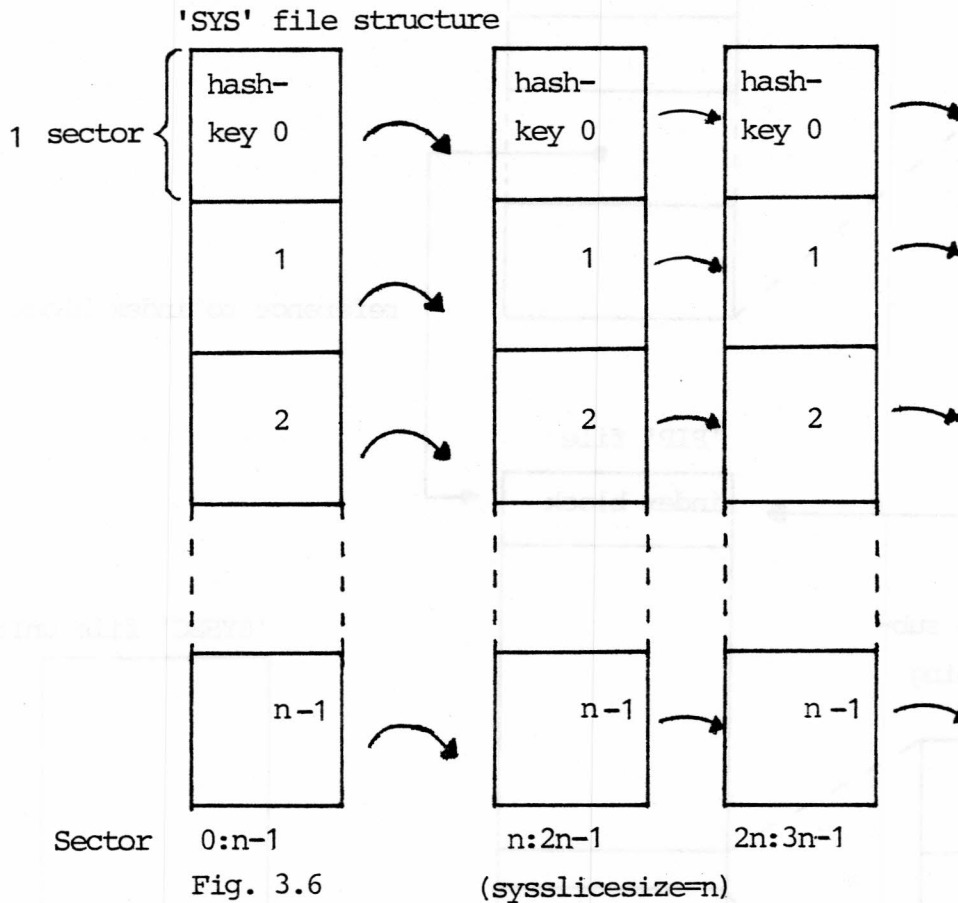
When a file is extended (with a slice) then the next slice description in the indexblock is used to keep the length of the slice and the sector number of the first sector in the slice. If however, the new slice is (physically) next to the previous slice, only the length in the slice description for the previous slice is updated. (see fig. 3.4.).

3.4. Catalog Format

The catalog is organized as an ordinary file except for the indexblock which is written in sector 6. The name of the catalog file is 'SYS'.

Each sector of 'SYS' contains 16 catalog entries which occupies 16 words each. An unused catalog entry is set to binary zero. Used catalog entries are accessed by name. Names are assigned to one of the sectors by hashing. The initial length of the

'SYS' file is defined in the unit description block (see fig. 3.3). If the number of catalog entries with equal hashkey exceeds 16, the catalog file is extended with a number of slices given as initial 'SYS' length. The catalog structure is shown in fig. 3.6.



3.5. Sub Catalog Format

3.5.

A sub catalog is an ordinary file (described in the catalog 'SYS' file) organized as the 'SYS' file. While the main catalogs are accessed by means of the 'CAT' process, sub catalogs are accessed by catalog processes with names equal to the sub catalog names. Catalog processes are created (and removed) by the 'CAT' process. A special file named 'SYSSC' contains a number of links to the sub catalogs accessible at the given time. Only the sub catalogs which is described by links in the 'SYSSC' file on unit 0, can be accessed. The administration of the 'SYSSC' file is done by a DOMUS utility program 'SUBCA'. Fig. 3.7 shows a sub catalog and the 'SYSSC' link to the sub catalog.

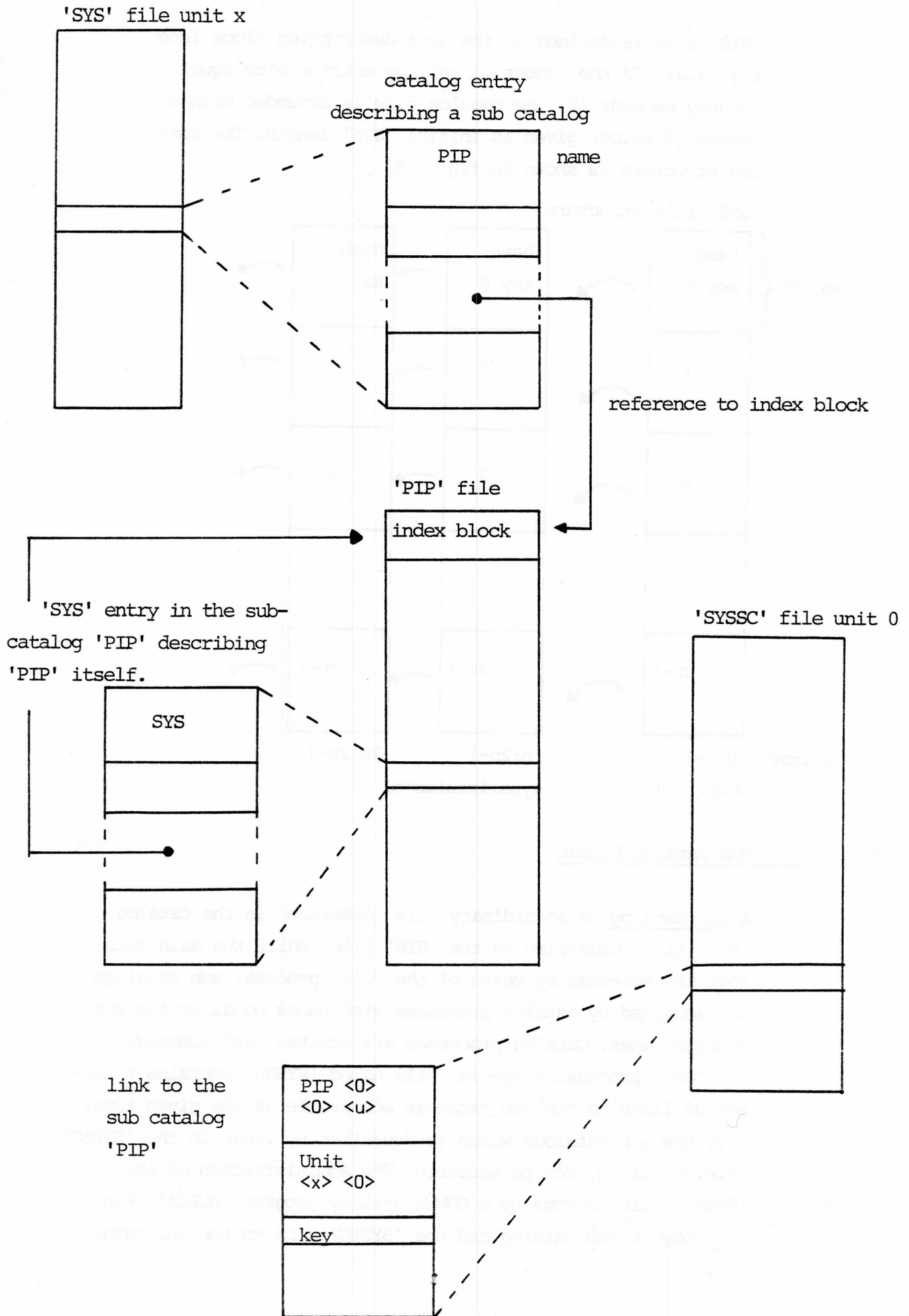


Fig. 3.7

3.6 Catalog Entry Format

3.6

A catalog entry consists of 16 words with the following format:

word	
0	
1	name
2	
3	
4	optional 1
5	
6	attributes
7	file length
8	index block sector
9	reserved length
10	
11	
12	tail
13	(optional 2)
14	
15

- 1) name: consists of 5 characters
- 2) optional 1 = binary zero initial
- 3) attributes: a number of bits describing the attributes of the file

bit 0: catalog file, i.e. 'SYS' or 'MAP' files

bit 1: sub catalog, the file is a sub catalog

bit 9: reserved

bit 10: reserved

bit 11: permanent, the file is permanent, i.e. the file cannot be removed or changed. Only the attributes can be changed.

bit 12: write protected, only reading is allowed on the file.

bit 13: entry only, reserved length = 0, no slices can be assigned for the file.

bit 14: reserved

bit 15: extendable, if bit 15 = 1 the file is extendable, else the file length is fixed.

- 4) file length: indicates the actual number of sectors occupied by the file.
- 5) index block sector: the sector number of the index-block for the file
- 6) reserved length: indicates the total number of sectors assigned for the file, i.e. reserved length \geq file length.
- 7) tail (optional 2): binary zero initial.

3.7. The 'SYSSC' File

A file named 'SYSSC' is used to describe a set of sub catalogs. Any unit may contain a 'SYSSC' file but only the set of sub catalogs described by the 'SYSSC' file on unit 0 can be accessed.

The organisation of the 'SYSSC' file is:

- a number of 16-word entries
- each entry is able to describe one subcatalog
- the entries are numbered from n to 255, where n is a parameter to the DOMUS utility program which can work on the 'SYSSC' file (DOMUS utility SUBCA).
- each used entry contains information about the sub catalog name, the unit on which the sub catalog is located and the sub catalog key.

Fig. 3.8. shows a part of the 'SYSSC' file.

In fact the entries in 'SYSSC' could be regarded as links to sub catalogs

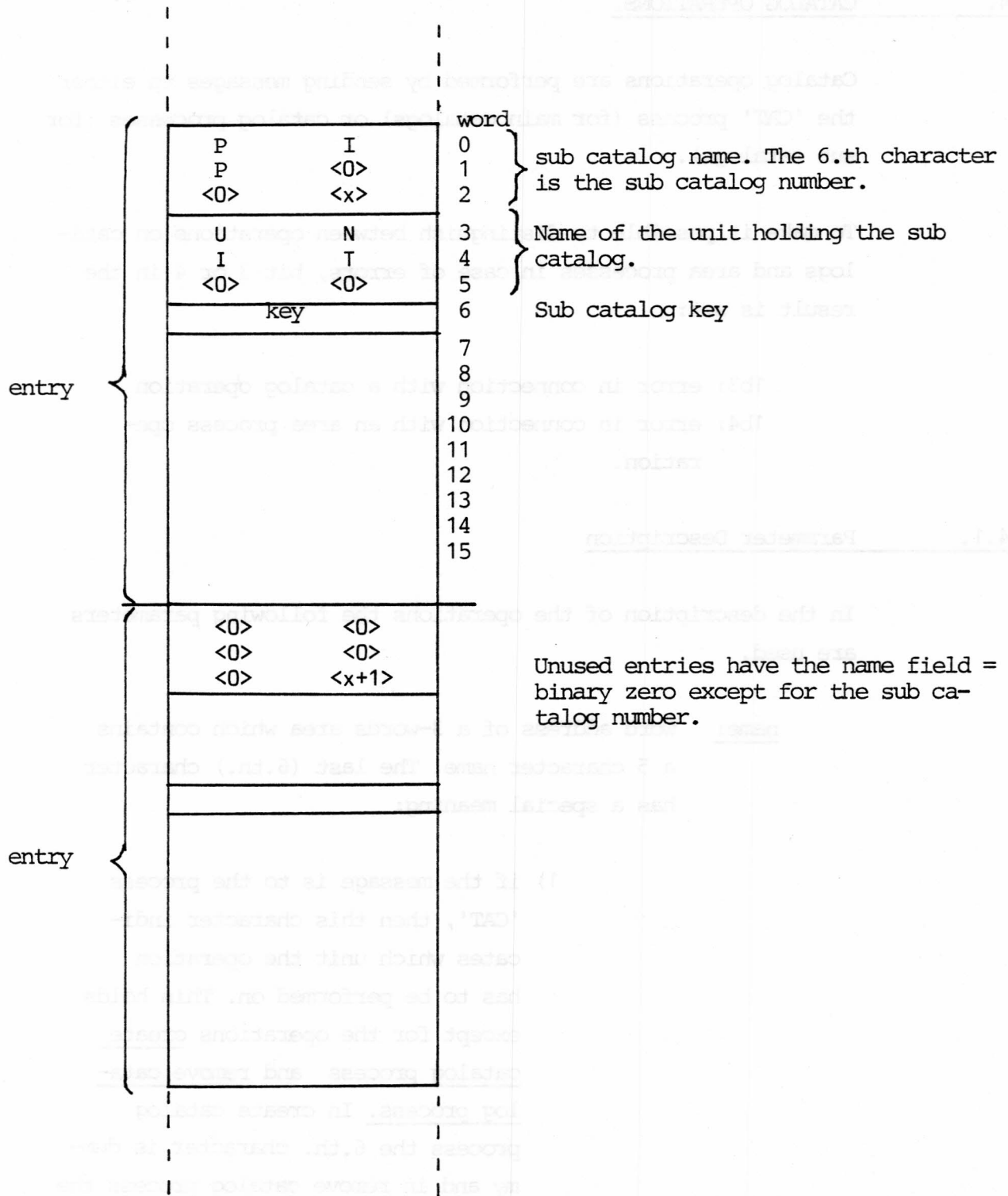


Fig. 3.8

4. CATALOG OPERATIONS.

4.

Catalog operations are performed by sending messages to either the 'CAT' process (for main catalogs) or catalog processes (for sub catalogs).

To make it possible to distinguish between operations on catalogs and area processes in case of errors, bit 3 or 4 in the result is set:

1b3: error in connection with a catalog operation

1b4: error in connection with an area process operation.

4.1. Parameter Description

In the description of the operations the following parameters are used.

name: word address of a 3-words area which contains a 5 character name. The last (6.th.) character has a special meaning:

- 1) if the message is to the process 'CAT', then this character indicates which unit the operation has to be performed on. This holds except for the operations create catalog process and remove catalog process. In create catalog process the 6.th. character is dummy and in remove catalog process the 6.th character must contain the sub catalog number (returned from the corresponding create catalog process operation).
- 2) if the message is to a catalog process the 6th character always contains the sub catalog number

(received from the create catalog process operation which created the catalog process).

entry word address of a 16-words area, formatted as a catalog entry.

If nothing else is mentioned, only buf. mess 0 (= result) in answers is relevant.

4.2. Main Catalogs

4.2.

Operations on main catalogs are handled by the process 'CAT'. The main catalogs (or units) are described in the separate process 'CATW'. The operations which can be performed on main catalogs are:

- 1) Set entry
- 2) Create entry
- 3) Look up entry
- 4) Change entry
- 5) Remove entry
- 6) Create area process
- 7) Remove area process
- 8) Init catalog
- 9) Create catalog process
- 10) Remove catalog process

4.3. Sub Catalogs

4.3.

Operations on sub catalogs are handled by catalog processes which have been created by the 'CAT' process. The name of a catalog process consists of the 5 character sub catalog name followed by (as the 6.th. character) a sub catalog number. Sub catalog numbers are gained from the file 'SYSSC<0>' which describes the sub catalogs which are accessible at the moment.

The operations which can be performed on sub catalogs are the same as for main catalogs except for 8,9 and 10 (see 4.2.).

4.4 Set Entry.

4.4

	Message
mess0	1b0
mess1	name
mess2	entry
mess3	not used

All fields in entry (except for entry.file-length and entry.indexblk) must be set by the calling process. A catalog entry is created with the content taken from entry. The number of slices necessary for entry.res-length is assigned to the catalog entry.

```

Result=      0, catalog entry created
Result<>     0, catalog entry is not created
1b3+ 1b0, catalog I/O error
1b3+ 1b6, parameter error, i.e.
           1) wrong size, entry.res-length < 0
           2) name format illegal
           3) illegal attribute
           4) disc not initialized
1b3+ 1b7, disc full
1b3+ 1b11,catalog entry with the same name already exists
1b3+ 1b12,indexblk full

Time=        3, disc accesses if entry.res-length = 0
            8, disc accesses if entry.res-length > 0

```

4.5 Create Entry.

4.5

	Message
mess0	1b1
mess1	name
mess2	size
mess3	attribute

A catalog-entry is created with:

```

entry.name = name
entry.optional = 0 (binary zero)
entry.attribute = attribute
entry.file-length = size
entry.tail = 0 (binary zero)

```

The number of slices necessary for size is assigned to the catalog entry.

```

Result=      0, catalog entry created
Result<>     0, catalog entry is not created
1b3+        1b0, catalog I/O error
1b3+        1b6, parameter error
              1) wrong size, size < 0
              2) name format illegal
              3) illegal attribute
              4) disc not initialized
1b3+        1b7, disc full
1b3+        1b11, catalog entry with the same name already exists
1b3+        1b12, indexblk full

Time=       3, disc accesses if size = 0
           8, disc accesses if size > 0

```

4.6 Look-up Entry.

4.6

	Message
mess0	1b2
mess1	name
mess2	entry
mess3	not used

The catalog-entry given by name is copied into entry.

Result= 0, entry = content of catalog-entry

Result<> 0, no entry, entry not found

1b3+ 1b0, catalog I/O error

1b3+ 1b1, catalog-entry with the given name does not exists

1b3+ 1b6, disc not initialized

Time= 2, disc accesses

4.7 Change Entry.

4.7

	Message
mess0	1b3
mess1	name
mess2	entry
mess3	not used

The catalog-entry given by name are changed in accordance with entry.

entry.name = new file-name.

If new file-name (0:1)=0 then catalog-entry.name is not changed.

entry.attribute = new attribute.

If new attribute < 0 then catalog-entry.attribute is not changed.

entry.res-length = new file-length.

If new file-length < 0 then catalog-entry.file-length is not changed.

If catalog-entry.attribute = permanent then only attribute can be changed.

Entry.optional and entry.tail gives the new content of catalog-entry.optional and catalog-entry.tail.

Result= 0, catalog-entry changed

Result<> 0, catalog-entry not changed

1b3+ 1b0, catalog I/O error

1b3+ 1b1, catalog-entry with the given name does not exist

1b3+ 1b6, parameter error, i.e.

- 1) change of name or length for a permanent file
- 2) illegal attribute
- 3) disc not initialized
- 4) an area process exists on the file

1b3+ 1b7, disc full
 1b3+ 1b11, catalog-entry with new file-name already exists
 1b3+ 1b12, indexblk full

Time in disc accesses=

a: change attribute : 3
 b: change name : 7
 c: change length :
 If new length > old length, then 8
 If old length = 0, then 7
 If new length < old length, then 8

Combinations: a+b = b, a+c = c, b+c = b+c, a+b+c = b+c.

4.8 Remove Entry.

	Message
mess0	1b4
mess1	name
mess2	not used
mess3	not used

The catalog-entry given by name is removed from the catalog.
 The slices (if any) assigned to the catalog-entry are released.

Result= 0, catalog-entry removed
 Result<> 0, not removed
 1b3+ 1b0, catalog I/O error
 1b3+ 1b1, catalog-entry with the given name does not exists
 1b3+ 1b6, parameter error, i.e.

- 1) disc not initialized
- 2) an area process exist on the file
- 3) the file is permanent

Time= 3, disc accesses if file-length = 0
 6, disc accesses if file-length > 0

4.9 Create Area Process.

4.9

	Message
mess0	1b5
mess1	name
mess2	not used
mess3	not used

If an area process with the given name already exists, the message is dummy. Otherwise an area process on the catalog-entry given by name is created. The area process descriptor is taken from the queue of free area process descriptors and initialized according to the catalog-entry (i.e. file-length, attributes). Then the area process is chained to the process-chain and started as a normal MUS process.

Result= 0, area process created or already exists
 1b4+ 1b0, catalog I/O error
 1b4+ 1b1, catalog-entry with the given name does not exist
 1b4+ 1b6, illegal, i.e.
 1) disc not initialized
 1b4+ 1b7, no area process descriptor available

Time= 0, disc accesses if an area process exists, else
 2, disc accesses

4.10 Remove Area Process.

4.10

	Message
mess0	1b6
mess1	name
mess2	not used
mess3	not used

If an area process with the given name does not exist, or the number of users of the area process > 0 then the message is dummy, else

- all messages in the event queue are returned,
- the area process is removed from process-chain,
- the catalog-entry associated with the area process is updated if there has been any change (i.e. file-length has been changed).

Result= 0, area process removed or area process does not exist or number of users > 0

1b4+ 1b0, catalog I/O error

1b4+ 1b6, illegal, i.e.

1) disc not initialized

Time= 0, disc accesses if number of users > 0 or area process does not exist, else

2, disc accesses

4.11 Init Catalog

4.11

	Message
mess 0	1b7
mess 1	not used
mess 2	unit no
mess 3	not used

The unit given by unit no is initialized. The unit description block (see fig. 3.3.) is read to get the information necessary to handle the unit.

Result =	0, catalog initialized
Result <>	0, catalog not initialized
1b3+	1b6, unit unknown
Time =	2, disc accesses

4.12 Create Catalog Process

4.12

	Message
mess 0	1b5+1b7
mess 1	name
mess 2	not used
mess 3	key

Key is an integer which indicates the usage of the sub catalog. Key = 0 means that only reading is to be performed on files in the sub catalog. Key <>0 means that any operation is to be performed on the sub catalog. If key <>0 then key is tested against the sub catalog key. If the test is ok then all operations are allowed. Note: the catalog system does not check the usage of the sub catalog except for the checking of keys in this operation.

The sub catalog with the given name is looked up in the 'SYSSC' file on unit 0. The entry in the 'SYSSC' file gives the unit on which the sub catalog is located. The entry also gives the actual sub catalog number which is inserted as the 6.th. character of the name.

If a catalog process with that name (inclusive the number) does not exist, the main catalog on the given unit is looked up for a catalog entry with the sub catalog name.

A process descriptor is taken from the queue of free area process descriptors and the catalog process is initialized according to the catalog entry which contains the file description for the sub catalog. It is chained to the process chain and started as a normal MUS process.

At return mess 2 contains the sub catalog number, which should be the 6.th. character of all name parameters to the catalog process in future use. Mess 3 contains the allowed usage. Mess 3 = 0 means that only reading should be performed, mess 3 <> 0 means that all operations are legal. It is the users responsibility that only operations allowed by mess 3 is performed.

If the catalog process already exists, then the message is dummy except for key (usage) checking.

Result=	0, catalog process created or already exist
Result <>	0, catalog process not created or illegal key
1b4+	1b0, catalog i/o error
1b4+	1b1, catalog entry does not exist. Either the 'SYSSC' file or the sub catalog file does not exist.
1b4+	1b6, illegal, i.e. <ol style="list-style-type: none"> 1) illegal key 2) disc not initialized, the unit containing the sub catalog is not initialized 3) the file is not a sub catalog
1b4+	1b7, no area process descriptor available for the catalog process

Time = 4 - n, disc accesses if the catalog process already exists. n is depending of the size of 'SYSSC' and the sub catalog number. (n<=19)

6 - n, disc accesses if the catalog process does not exist. (n<=21)

4.13 Remove Catalog Process

4.13

	Message
mess 0	1b6+1b7
mess 1	name
mess 2	not used
mess 3	not used

If any user of the catalog process exists or the catalog process has any not released area processes or the catalog process does not exist, the message is dummy, else

- all messages in the eventqueue are returned
- the catalog process is removed
- the catalog on the actual unit is updated if necessary

Result= 0, catalog process removed or catalog process does not exist or number of user >0 or number of area processes created by the catalog process >0

Result <> 0, catalog process not removed

1b4+ 1b6, illegal, i.e.

- 1) the process is not a catalog process

Time= 0, disc accesses

4.14 Get Unit Description

4.14

	Message
mess 0	1
mess 1	32
mess 2	byte addr
mess 3	unit no.

This message, which is a normal input message, will move the unit description given by unit no to the area pointed out by byte addr. The number of bytes moved is 32. The format of a unit description can be seen in section 7.1.

Result= 0, the unit description is delivered

Result <> 0, the unit description is not delivered

1b3+ 1b6, illegal, i.e.
 1) count (mess 1) < 32
 2) unit unknown

Time = 0, disc accesses

5. AREA PROCESSES

5.

For description of how to create and remove files on disc, initialize catalogs, and create and remove area processes see chapter 4.

The name of an area process consists of 6 characters. The first 5 correspond to the name of the disc file, and the 6th 'character' is the number of the unit or sub catalog on which the file is stored. This number should be binary.

The first block in a file has the relative number zero. Both control and transput messages are accepted.

5.1 Control

5.1

Reservation for use, exclusive user, and exclusive writer are accepted. Besides these only position is accepted.

Reservation: If it is possible, the calling process may be included as user, exclusive user, or exclusive writer, or the user may be removed.

At the most 3 different users are possible, and each user has an open/close count which follows the number of reservation commands to the user entry. (each user = one process)

Count = 2: Insert user as exclusive writer.
 All other processes are allowed to
 read the file.
 The open/close count is increased.
 If the user is new, the usercount is
 increased.

Result= 0: The reservation is accepted.
 1b6: Area process reserved for exclusive
 use by another process.
 1b4 + 1b6: The area process is already reserved
 for exclusive writing by another process
 1b4 + 1b12: No more user entries.

Count = 3: Insert user.
All other processes are allowed to read the file, and the area process may be reserved for exclusive writing too.
The open/close count is increased.
If the user is new, the usercount is increased.

Result = 0: The user is inserted as a user of the area process.
1b6: The area process is reserved for exclusive use by another process.
1b4 + 1b12: No more user entries.

Count = 4: Insert user as exclusive user.
Any other process is rejected.
The open/close count is increased.
The usercount can only be 1.

Result= 0: Reservation accepted.
1b6: Area process reserved for exclusive use by another process.
1b4 + 1b6: Area process is being used by another process.
1b4 + 1b12: No more user entry.

Count = 0: Remove user.
The open/close count is decreased. If it reaches zero, the usercount is decreased too, and if the user was either exclusive user or writer, this state is cleared.

Result = 0: The user is removed.
1b6: Area process reserved for exclusive use by another process.
1b4 + 1b11: The calling process is not a user of the area process.

Time = 0 disc accesses

Position: The position in mess 3.buf withing the file is set.

If the position < 0, then the position := 0

If the position > file size, then the position
:= file size.

Result = 0: Position accepted.

1b6: Area process reserved for exclusive
use by another process.

1b4 + 1b6: Position < 0

1b4 + 1b11: 1. Position is outside the file.

2. The sender is not a user of the
area process.

Time = 0 disc accesses.

Other control messages: The disc is sensed to check the status.

Result = 0: OK

1b6: Area process reserved for exclusive use
by another process.

1b4 + 1b11: The sender is not a user of the area
process.

Time = 0 disc accesses.

5.2 Transput

5.2

	Message	Answer
mess 0.buf	operation	status
mess 1.buf	bytecount	number of bytes
mess 2.buf	byteaddress (wordaddress)	unchanged
mess 3.buf	(first block No.)	first block No.

Bytecount in mess 1.buf must be 512 bytes.

Input

Operation = 1 Input sequential mode. Reading of the file continues at the position reached by the last transport or position message.

Mess.3.buf is irrelevant in the message, but in the answer it is set to the number of the block first read.

Operation = 5 Input random mode. Before reading the file is positioned to the position given in mess3.buf. Else as for operation = 1

Output

Operation = 3 Output sequential mode. Writing in the file continues at the position reached by the last transport or position message.

The file is automatically extended if attribute (extension possible) is set.

Mess3.buf is irrelevant in the message but in the answer it is set to the number of the block first written.

Operation = 7 Output random mode. Before writing the file is positioned to the position given in mess3.buf. Else as for operation = 3.

Operation = 19 Read after write in sequential mode. First an output operation = 3 is carried out. Then the written segment is read for check of status. For some disc deviced the output buffer is used as input buffer during the check read.

BIT	CAT76 + 1b3 or 1b4	RC 3652	RC 3650	RC 8221, 22, 23, 24, 25, 26
1b0	Catalog i/o error	Unit not available, malfunction (hardware (12)).	Disconnected, unable to accept commands.	Disconnected, power off, hard error, unit not available.
1b1	Entry does not exist	-	Off line	-
1b2	-	-	-	-
1b3	Message from CATHANDLER	-	-	-
1b4	Message from area process	-	-	-
1b5	-	-	Write-protected	Write-protected
1b6	Name format illegal, disc not initialized, attribute error, file length < 0, area process reserved (1b6 - only). EOF of fix length file (+1b4)	Driver reserved by another process	Driver reserved by another process	Driver reserved by another process
1b5+1b6	-	Unit does not exist	Write-protected in output	Unit does not exist, multiple or no unit selected, invalid drive command, write-protected in output.
1b7	Disc full (+1b3), no more area processes. (+1b4)	-	-	-
1b8	-	Block error	Block error	Block error
1b6+1b8	-	Odd byteaddress	-	Odd byteaddress
1b9	-	Data late	-	-
1b10	-	Parity error	Parity error	Parity error
1b11	Entry exists (+1b3), not user of area process (+1b4), position is outside the file (+1b4).	-	End of existing kit	End of existing disc kits
1b12	Index block full, no more user entries (+1b4).	Seek error (position error)	Hardware position error, out- side existing kit (+1b11).	Hardware position error
1b13	-	-	-	-
1b14	-	Time-out	Time-out	Time-out
1b15	-	-	-	-

) If nothing else is mentioned, one of these bits is added to the error bits in this column.

6. FORMAT OF PROCESS DESCRIPTIONS

6.

A process description (either for an area process or for a catalog process) occupies 74 words. The first 25 words correspond to a normal driver process description, see MUS Programmer's Guide. The remainder of the process description differs if it is an area process or a catalog process. The two formats are described in the following two sections.

6.1 Area Process Description

6.1

Area process descriptor

- +25 attribute
- +26 file length (= actual length <= reserved length)
- +27 address of index block
- +28 reserved length (= total number of sectors assigned for the file)
- +29 work location
- +30 address of the unit description for the unit which contains the file
- + (31:38) work locations
- +39 next position entry (= next victim when a position is not found among the current user positions)
- + (40:42) user position 1. Contents:
 - + 40 sector number of the slice
 - + 41 first block number rel. to file start
 - + 42 top block number rel. to file start
- + (43:45) user position 2. contents as for position 1.
- + (46:48) User position 3. contents as for position 1.
- +49 work location

- +50 user count (= number of users of the area process)
- +51 exclusive writer (process descriptor address of the exclusive writer or 0)
- +52 work location
- + (53:55) user entry 1. Contents:
 - +53 user identification (= process descriptor address of user)
 - +54 user position rel. to file start (after a transport, mess3= user position)
 - +55 open/close count
- + (56:58) user entry 2. Contents as for user entry 1.
- + (59:61) user entry 3. Contents as for user entry 1.
- +62 work location
- + (63:72) message buffer
- +73 work location

6.2 Catalog Process Description

Catalog process descriptor

- + (25:29) work locations
- +30 address of the unit description for the unit which contains the sub catalog.
- + (31:32) work locations
- +33 address of index block for the sub catalog
- + (34:39) work locations
- +35 number of active area processes created by the catalog process.
- +36 number of users of the catalog process. Each create catalog process operation increases this count and each remove catalog operation decreases it.
- + (37:62) work locations
- + (63:72) message buffer
- +73 work location

7. OPTION PROGRAMS

7.

In connection with the catalog system a number of option programs exists. Already mentioned are the option programs which contains area processes. The names of these programs are 'CAPX' where X may be 2, 3, 4 or 8 giving the number of area processes included in the program. Any number (and combination) of the 'CAP' programs may be loaded during system run.

Every system must contain a 'CATW' process describing the units (main catalogs) which are accessible. The format of the 'CATW' process and how to produce another 'CATW' process is described in the next section.

7.1 Producing 'CATW'

7.1

The format of 'CATW' is:

- word 0 program specification (=1b0+1b1+1b7+1)
- 1 start of program
- 2 program chain (set by the MUS system)
- 3 size of program
- 4:6 name of program (='CATW<0><0>')
- 7 first area process
- 8 top area process = first area process

: head of unit description chain

- 9 head of unit chain
- 10 chain of head of unit chain
- 11:30 first unit description
- 31:50 second unit description

:

:

: in the last unit description the chain field must be zero.

word 10+n*20 : n unit descriptions, followed by
 a process description (process name
 = 'CATW<0><0>').

The process description is followed
 by a piece of code which inserts the
 'CATW' process and the code as an area
 process in the area process chain, which
 in turn means that the 'CATW' process re-
 moved (the 'CATW' program is still in
 the program chain).

The format of a unit description is:

word 0	reference to driver name
1	unit number (binary)
2	chain to next unit
3	size of unit description
4:6	name of unit (i.e. 'UNIT<1><0>')
7:9	name of driver for the unit.
10:11	kit displacement = number of sectors prior to this unit, which is handled by the driver.
12	'SYS' size
13	slice size
14	number of sectors on unit
15	number of free sectors on unit, init. value = number of data sectors =(number of sectors on unit) - (first data sector). Set by the user, changed by 'CAT' during run.
16	first data sector, points to the catalog file for the unit. Constant set by the user.
17	top data sector, point to the last sector +1 of the unit = number of sectors on unit. Con- stant set by the user.
18:19	not used

Word 0:11 is defined when the 'CATW' program is produced while word 12:17 (12:19) is set when the unit is initialized.

If a new 'CATW' program has to be produced then use a program named CATW1 as a source and replace the unit descriptions in this program with the new descriptions.

7.2 Init New Catalog

7.2

When a new disc should be used by the catalog system it must first be initialized by a special option program named 'CATT'.

Initialization is done by sending the following output message to 'CATT':

	Message
mess0	1b8+3
mess1	18 (bytecount)
mess2	byteaddress
mess3	not used

The byteaddress must point to an 18-byte record with the content:

<u>word</u> 0	unit number of the unit to be initialized
1:8	unit parameters as described in section 7.1, unit description words (12:19)

Note: If there already exists a catalog on the disc then the information contained within this catalog is lost.

The 'CATI' process will only accept the commands: reservation and output with 1b8. All other commands (except for sense) are illegal.

Result= 0: unit initialized
1b6: the process is reserved
1b3+1b6: the unit description is not found
1b3+1b6: illegal command
1b3+1b8: byte count <>18

Time= 8+n, where n= 'SYS' size.

