A/S REGNECENTRALEN

SCANDINAVIAN INFORMATION PROCESSING SYSTEMS

RCSL No: 55-D145
Edition: January 1971
Author: Hans Rischel

Title: How to Make Autoload Magnetic Tapes for RC 4000

Keywords: RC 4000, Basic Software, Autoload, Report

Abstract: Detailed operators instructions are given for the use of the program CREATE (RCSL 55-D76) for making an autoload magnetic tape. 5 pages.

DK-2500 VALBY · BJERREGAARDSVEJ 5 · TELEPHONE: (01) 46 08 88 · TELEX: 64 64 rcinf dk · CABLES: INFOCENTRALEN

How to Make Autoload Magnetic Tapes for RC 4000

On RC 4000 installations with magnetic tape stations a magnetic tape with monitor and system programs for fast initialization of the computer can be made by means of the program 'create'. Besides the magnetic tape 'create' produces two small paper tapes marked 'autoload monitor' and 'autoload system'

The autoloading is done by loading one of the paper tapes by means of the autoload button on the operator panel (like a monitor on paper tape). The console will now output

input device xx

of which xx is the number of the device which was used for making the system tape. NL having been typed, the loading of the monitor will start from device xx. If the paper tape marked 'autoload monitor' is used, the operation will continue with 'oldcat' - i.e. the catalog on the backing store is retained - and the loading of the monitor is finished with the output

old catalog
from s
ready

and the computer is ready for the newdate command and the creation of processes. If the paper tape marked 'autoload system' is used, the 'initialize catalog' program will automatically start reading from the next file on the magnetic tape when the monitor has been loaded.

When the new catalog has been loaded, the computer is ready. During the operation we get the console output

new catalog

and finally

from s
ready

As it appears from the description above, each paper tape loader corresponds to a certain device No. and a certain file No. on the magnetic tape.

If when the console outputs

input device xx

we type a number yy and then NL, the machine will henceforward read from device No. yy. This will only work out well if device xx and device yy are of the same kind (both 7-track or both 9-track).

The creation of a system tape with monitor 2 on file 0 and various programs on file 1 (file processor, utility programs, compilers, etc.) can be performed in the following way: We create a process with pk 0 and pr 0 ('create' can run only in monitor mode), mount the tape, call it (for example) sysmt, and then perform the following fp-commands:

```
monitorfile =set mto sysmt
sysfile     =set mto sysmt 0 1
monitor2    =set 40
monitor2    =binin tro.1.1
```

We now load the paper tape monitor 2. When this binin call is used, the loader at the beginning of the tape is skipped, and only the monitor itself is loaded. (This will not work with monitors made before December, 1970).

```
lookup create
if ok.no
(message load create
binin tro)
monitorfile=create monitor2
```

The tape which now is punched should be loaded into the reader for check-reading. It is divided in two parts; the first half is marked 'autoload monitor 2 from device no. xx', the second half 'autoload system 2 from device no. xx'.

The programs on the next file are easily produced by means of binin-binout; please note, that 'newcat' should come first and 'end' at the very end. Here you may profit by the fact that the file processor tape starts with 'newcat' and ends with 'end' by

```
area1=set 100
message load fileprocessor
area1=binin tro.s.36
```

The number 36 is 1 less the number of binary segments on the actual file-processor tape (RCSL No. 55-T84), for which reason everything except the final 'end' is included in area 1. We make our 'end' for later use

```
endarea=set 1
endarea=slang
s. w. <:end:> e.
```

The programs wanted as backing areas can now be loaded at will according to the model

```
area2=set 100
message load fp utility
area2=binin tro.s
area3=set 100
message load algol compiler
area3=binin tro.s
```

and so forth (100 segments is ample - and yet boss needs a bit more than 250 segments).

The system file itself is written with a single call of binout:

```
sysfile=binout area1.ne.a area2.ne.a ...,
          ...     endarea.ne.b
```

EXAMPLE:

```
;production of autoload system magnetic tape
;to be loaded by i trf

clear sysstart
r=set
r=edit
i A

;sysstart

message did you remember pk 0 and pr 0
lookup create
if ok.no
(message load binary create program tape
binin tro)
message load binary monitor 2
monitor2   =set 40
monitor2   =binin tro.1.1

area1       =set 100
area2       =set 100
area3       =set 100
area4       =set 100
area5       =set 100
```

```
area6       =set 100
area7       =set 100
area8       =set 500
if ok.no
(message no room on bs
finis)
endarea     =set 1
endarea     =slang
s. w. <:end:> e.
message load binary file processor tape
area1       =binin tro.s.36
message load binary fp utility tape
area2       =binin tro.s
message load binary algol compiler tape
area3       =binin tro.s
message load binary algol library tape
area4       =binin tro.s
message load mts binary tape
area5       =binin tro.s
message load tapesort binary tape
area6       =binin tro.s
message load div binary tape
area7       =binin tro.s
bossfile    =set mto bossmt 0 1
area8       =binin bossfile.s

i sysout

A,f
rename r.sysstart

clear sysout
r=set
r=edit
i A

;sysout

monitorfil =set mto sysmt 0 0
systemfil  =set mto sysmt 0 1

monitorfil =create monitor2
systemfil  =binout area1.ne.a area2.ne.a area3.ne.a,
                   area4.ne.a area5.ne.a area6.ne.a,
                   area7.ne.a area8.ne.a endarea.ne.b
message end prod of autoload mag tape
end

A,f
rename r.sysout
r=set
(end
i sysstart)
```