

## 4 Monitor Modules: Flowcharts and Functional Description

	Page
INIMON . . . . .	2-72
D:LDER . . . . .	2-74
M:DISP . . . . .	2-76
I:PFAR . . . . .	2-90
I:RTC . . . . .	2-94
I:LKM . . . . .	2-98
M:IORM . . . . .	2-102
M:WAIT . . . . .	2-106
M:EXIT . . . . .	2-108
M:GBUF, M:FBUF . . . . .	2-113
D:CNFM . . . . .	2-116
D:DNFM . . . . .	2-120
M:ACT . . . . .	2-122
M:SWTC . . . . .	2-124
D:ATDT . . . . .	2-126
D:GTIM . . . . .	2-130
M:RSEV . . . . .	2-132
D:CNLV . . . . .	2-134
D:DNLV . . . . .	2-136
D:WGT . . . . .	2-138
ASGPRO . . . . .	2-140
DELFRO . . . . .	2-147
KEYPRO . . . . .	2-150
CANKEY . . . . .	2-152
D:SWP . . . . .	2-154
M:DFM . . . . .	2-156
D:CTPN, D:OCOM . . . . .	2-158
M:DMA . . . . .	2-162
M:DNL . . . . .	2-164
M:DCK . . . . .	2-166
F:BLK . . . . .	2-172
M:AOO . . . . .	2-174
I:TRAP . . . . .	2-178

## NUCLEUS INITIALIZATION PROGRAM (INIMON)

### Calling Sequence

This module starts the monitor when it has just been loaded from disc. Therefore it is entered from the IPL without any parameters.

### Work Areas and Tables

INIMON initializes:

- the disc tables: T:FCT, DWT, T:DCT
- memory partitioning: CMT
- the program control tables: T:PCT.

Besides, it uses one buffer (CATBUF, 205 words) to read the directory from disc.

### Input/Output File

The INIMON module outputs messages on the typewriter, if necessary, and reads the directory from disc. It also activates D:USV3 which will create the D:CI file.

Callout Messages

The loader is activated as any other software level program by the dispatcher or the command language. Upon entry, register A must be set to the PCT of the program which must be loaded. Therefore the PCT must already have been initialized by the command language before the loader is activated.

The loader requires about 250 words for the dynamic allocation area to read the program which is loaded. As soon as loading has been completed, this work area is updated. This work area is not allocated for any program (read only and swap programs) and is updated.

Work Areas and Work Areas are updated. This work area is updated. This work area is updated.

Input/Output Files

The loader has to use the dynamic allocation area to load the program. for Disc File Management is used in this system. The input file of the loader is therefore the file code (from /FO to /F7).

Error messages are output via the error typewriter (file code /E).

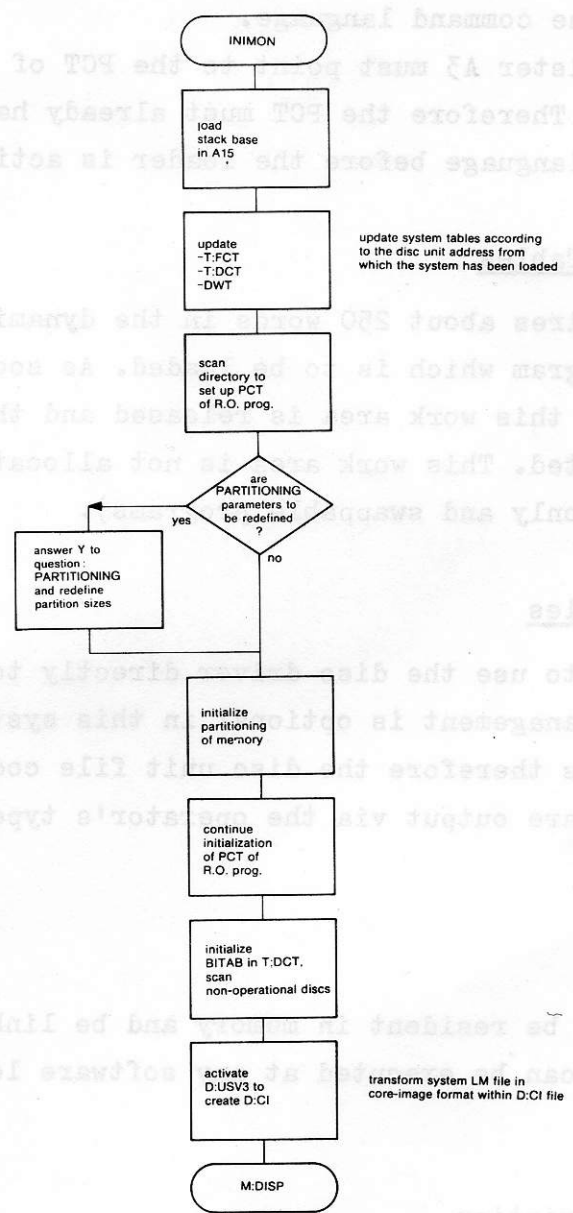
Memory Layout

The loader must be resident in memory and be link-edited with the supervisor. If one of the software levels, without re-activation.

Function Description

The principal function of the loader is to load a program into disc into memory. The program must be stored on the disc in the standard load module format.

In memory, the program can be stored in any area: Memory Resident, Read Only, Swap or Background Area. The program's PCT must have been initialized before the loader is called. The loader therefore



## LOADER (D:LDER)

### Calling Sequence

The loader is activated as any other software level program by the dispatcher or the command language.

Upon entry, register A3 must point to the PCT of the program which must be loaded. Therefore the PCT must already have been initialized by the command language before the loader is activated.

### Work Areas and Tables

The loader requires about 250 words in the dynamic allocation area to read the program which is to be loaded. As soon as loading has been completed, this work area is released and the PCT of the program is updated. This work area is not allocated for core image programs (read only and swappable programs).

### Input/Output Files

The loader has to use the disc driver directly to load the program, for Disc File Management is optional in this system. The input file of the loader is therefore the disc unit file code (from /FO to /FF). Error messages are output via the operator's typewriter (file code /EF).

### Memory Layout

The loader must be resident in memory and be link-edited with the supervisor. It can be executed at any software level, without restriction.

### Functional Description

The principal function of the loader is to load a program from disc into memory. The program must be stored on the disc in the standard load module format.

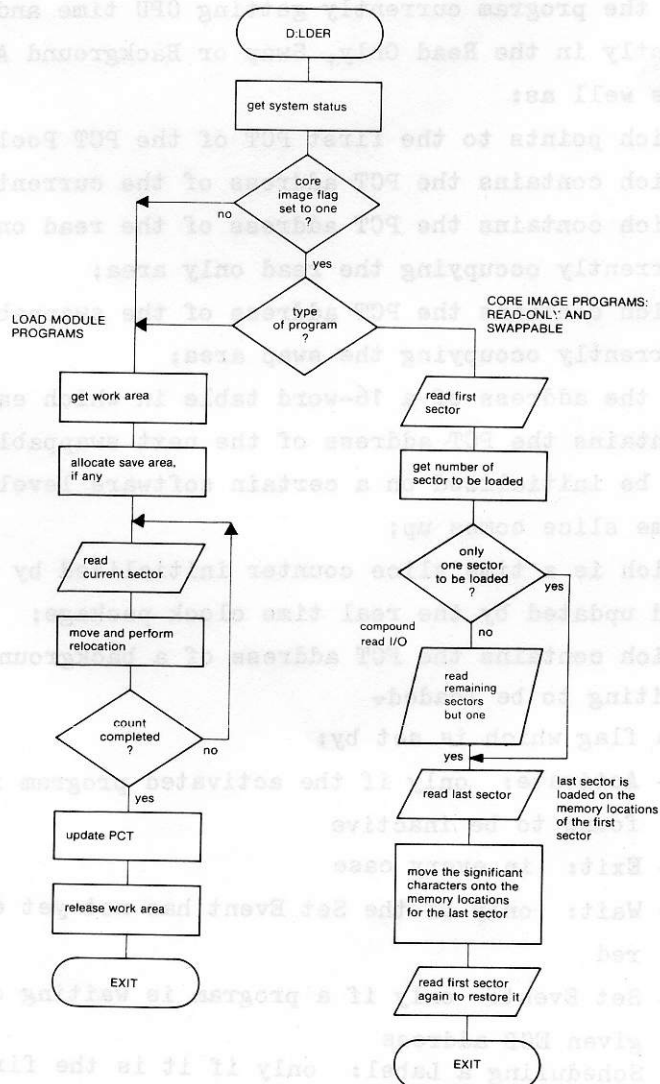
In memory, the program can be stored in any area: Memory Resident, Read Only, Swap or Background Area. The program's PCT must have been initialized before the loader is called. The loader therefore

does not have to search for the program on disc, but read the sector GRANTB (Granule Table) of the load module file, to find the absolute sector number where it must start reading. Having read a sector of the program into the dynamic allocation area, the loader will move instructions and data from the dynamic area to the area where the program is stored. Relocation is done at the same time. On completion of the loading operation, the PCT is updated and the buffer in the dynamic allocation area is released. The function of the Core Image Loader is to load a program from the D:CI file. The program is stored on disc in core image format.

Error Messages

In case of a disc I/O error during loading, one of the following messages is output:

- during loading of a load module:  
LDER <program name>
- during loading of a core image module:  
CI ER <program name>



## M:DISP (DISPATCHER)

### Calling Sequence

A6: Scheduled Label address, if a scheduled label is to be dispatched. If not: 0.

A5: Address of the PCT entry to which the scheduled label applies. (Irrelevant if A6 = 0).

ABL M:DISP

The dispatcher can also be called indirectly, via the third word of the Communication Vector Table; this allows a program, which has been link-edited separately from the dispatcher, to branch to it.

It is assumed that the A15 stack contains P-register, Program Status Word and registers A1 to A8 of the interrupted program.

### Work Areas and Tables

- A15 stack, to save and restore the program context.
- T:SLT (Software Level Table) } of the program it has to start
- T:PCT (Program Control Table) }
- Pointers to the program currently getting CPU time and the programs currently in the Read Only, Swap or Background Area are modified, as well as:
  - T:PCT , which points to the first PCT of the PCT Pool;
  - P:CUR , which contains the PCT address of the current program;
  - P:ROPC , which contains the PCT address of the read only program currently occupying the read only area;
  - P:SWAP , which contains the PCT address of the swappable program currently occupying the swap area;
  - T:SWP , is the address of a 16-word table in which each word contains the PCT address of the next swappable program to be initialized on a certain software level when its time slice comes up;
  - CVTTSC , which is a time slice counter initialized by the loader and updated by the real time clock package;
  - D:PCTB , which contains the PCT address of a background program waiting to be loaded.
- F:DISP , a flag which is set by:
  - Activate: only if the activated program is found to be inactive
  - Exit: in every case
  - Wait: only if the Set Event has not yet occurred
  - Set Event: only if a program is waiting on the given ECB address
  - Scheduling a Label: only if it is the first



- one for the current program
- Switch inside Level: only if the request is  
fective, i.e. if another PCT waiting for CPU  
time is found waiting on that level.

### Input/Output Files

None.

### Memory Layout

M:DISP is always resident in memory and belongs to the supervisor part of the DRTM.

### Functional Description

The dispatcher is called by any system routine running at a level equal to or below 48, especially when an event, such as I/O, has occurred and a scheduled label must be started.

Its main function is to determine which program must get CPU time. If the interrupted program was running at a level equal to or below 48, control is returned to it immediately. Then the F:DISP flag is checked: if it is not set, control is returned also to the interrupted program. Otherwise it will scan the Software Level Tables (T:SLT) and Program Control Table (T:PCT) to find the highest priority active program. If this program is already in memory, it will restore the registers from the stack and start it immediately. If it is not in memory, M:DISP will activate the loader to load the program, and look for the highest priority program.

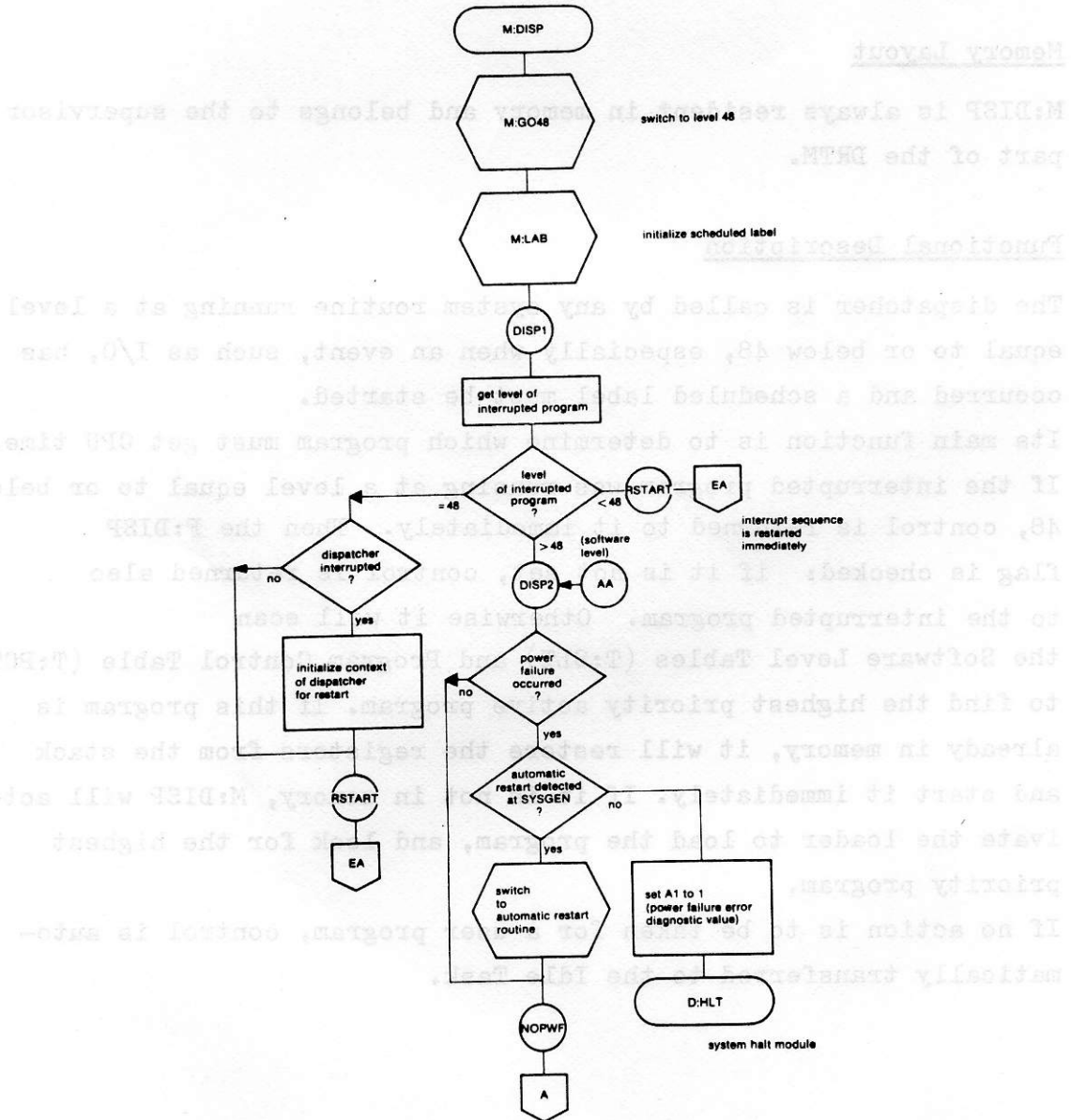
If no action is to be taken for a user program, control is automatically transferred to the Idle Task.

one for the current program  
 - Switch inside level: only if the request is  
 active, i.e. if another PCT waiting for CPU  
 time is found waiting on that level.

Input/output files

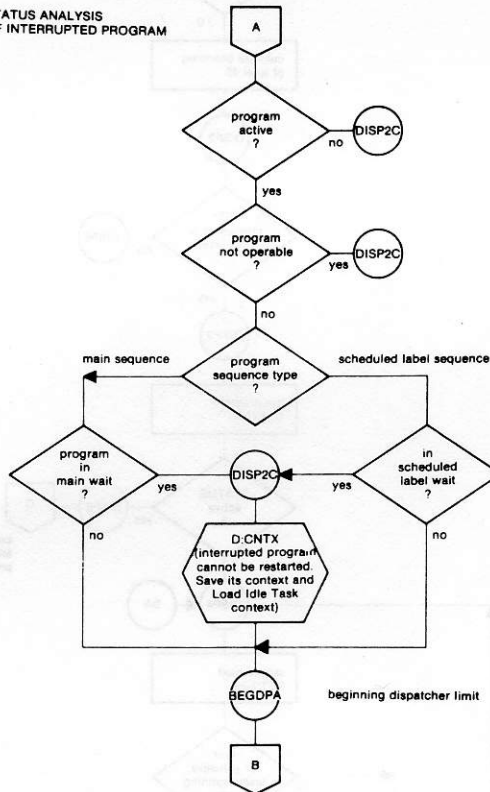
None.

Memory layout

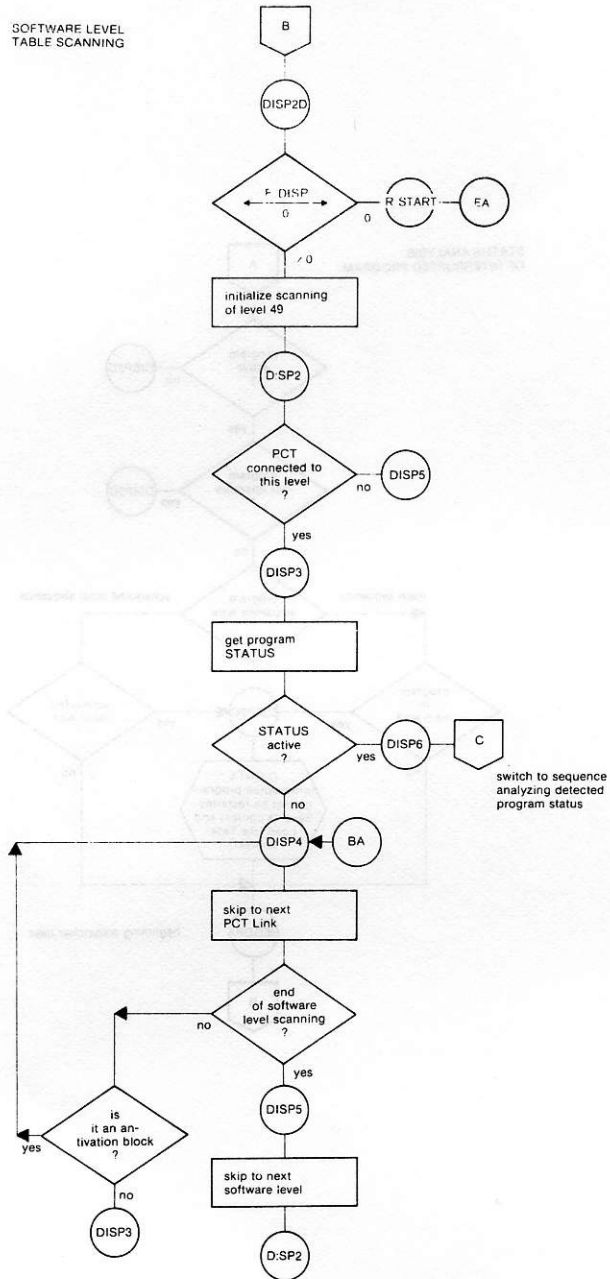




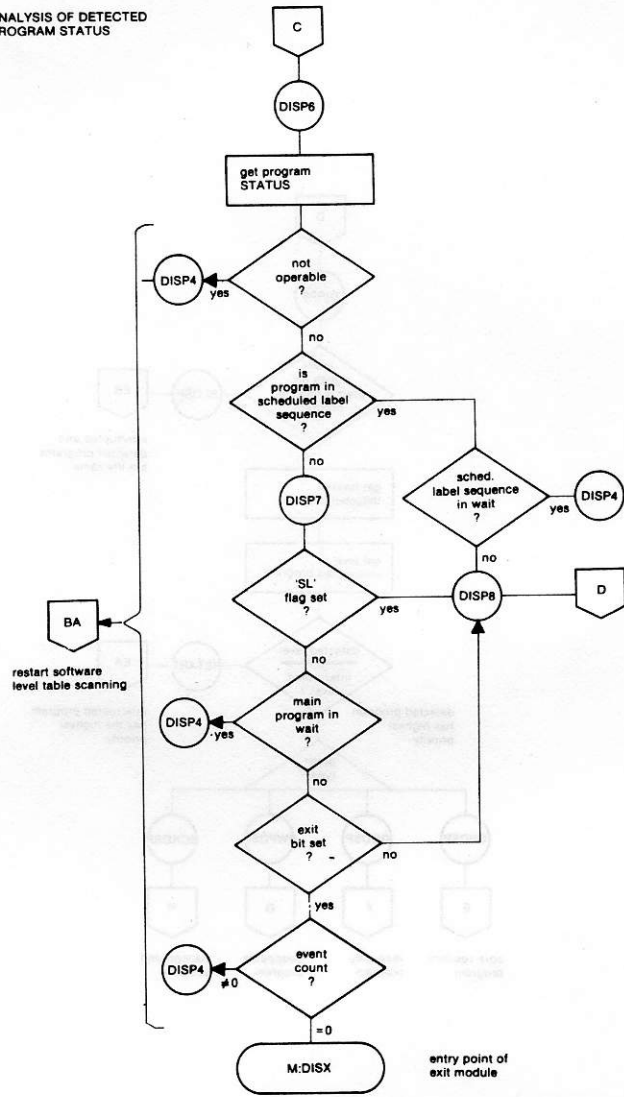
STATUS ANALYSIS  
OF INTERRUPTED PROGRAM

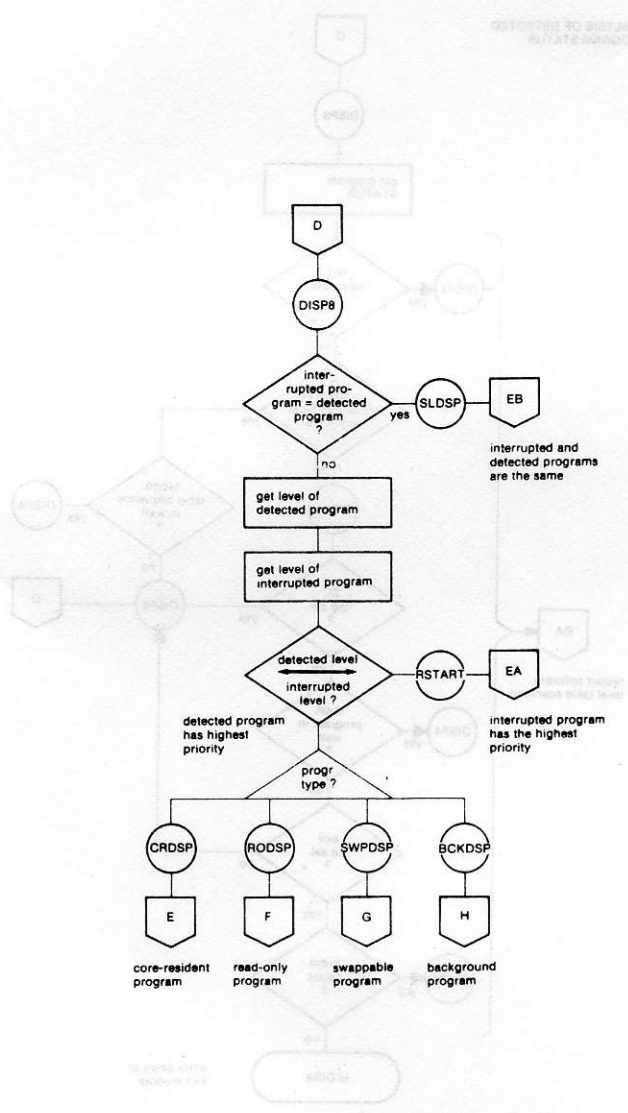


SOFTWARE LEVEL  
TABLE SCANNING

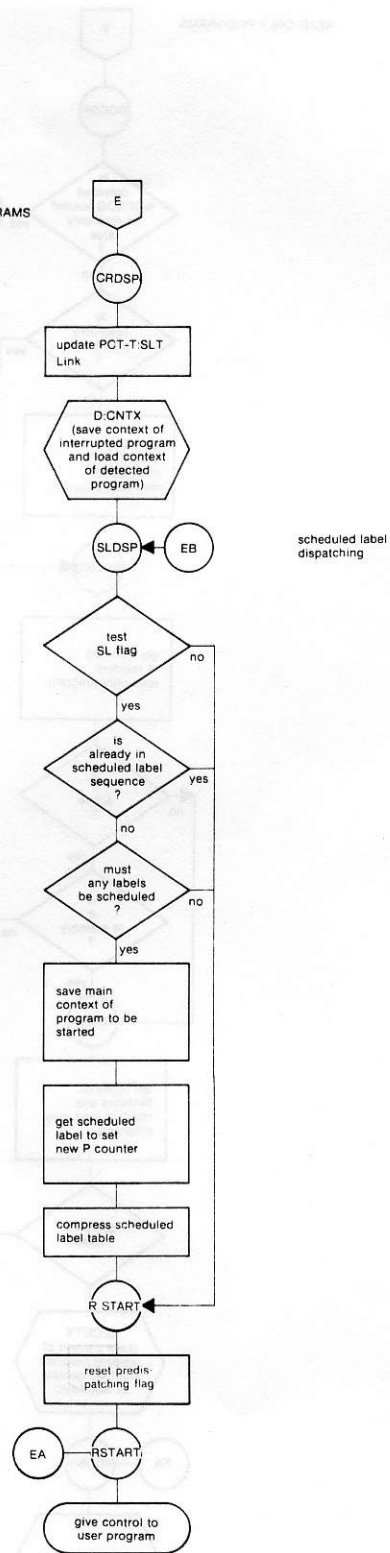


ANALYSIS OF DETECTED PROGRAM STATUS



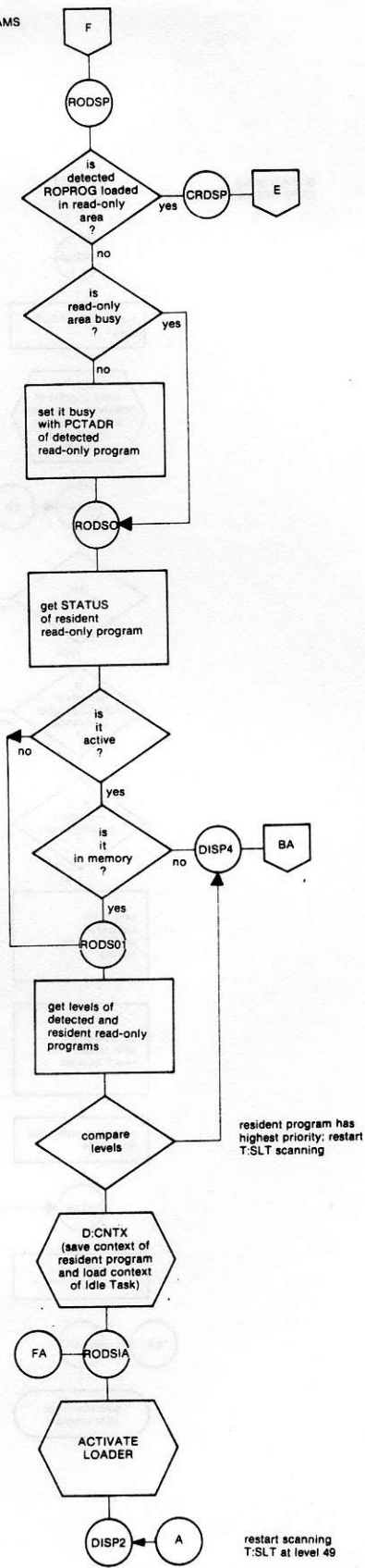


DISPATCHING  
CORE-RESIDET PROGRAMS

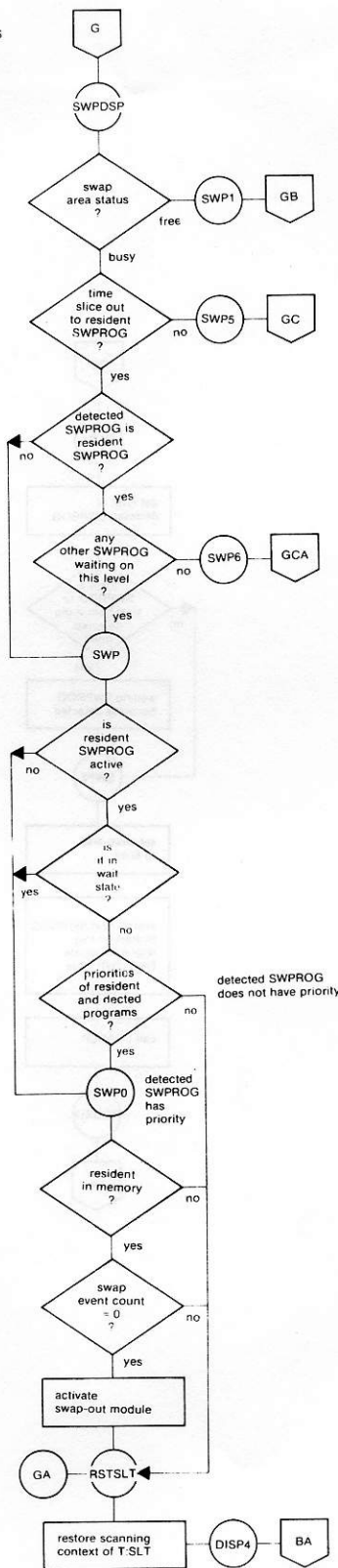


scheduled label  
dispatching

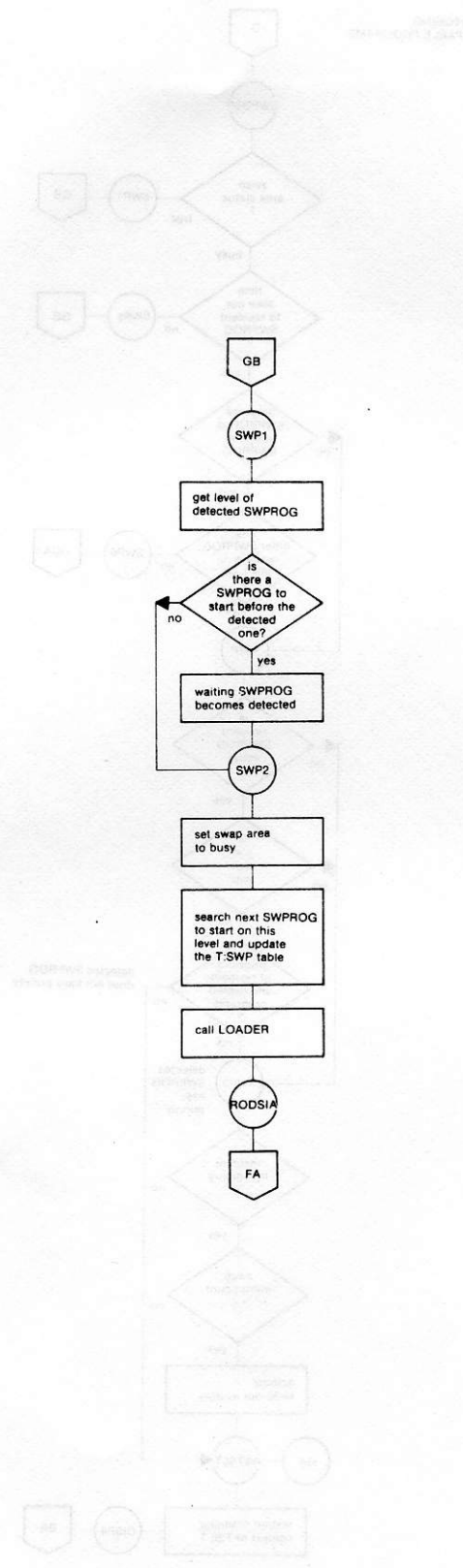
READ-ONLY PROGRAMS

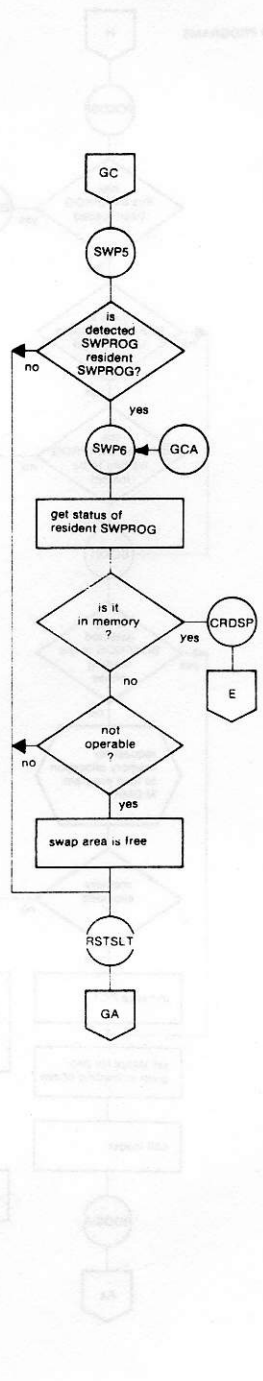


DISPATCHING  
SWAPPABLE PROGRAMS









DISPATCHING  
BACKGROUND PROGRAMS

