

## ASSEMBLY OF THE INITIAL CORE LOAD

The operating system software is initialized beginning with an initial core load (ICL). This document describes the steps necessary for the production of the ICL. We assume the existence of a number of source files having more or less obvious names. It is not proper to use their exact names or mention their whereabouts in this document, however, as experience has shown that they change names and locations too often to incorporate permanently into instructions such as these.

The actual ICL is produced with a simple 940 program called "MAKE-ICL". This program exists only in source, so it must be assembled with NARP and loaded with DDT before it can be run. Experience has shown that it is a good idea to list out the source before assembling it to see if the source is compatible with that software being placed into the ICL. Usually changes in the software dictate relocating some pages or changing the number of pages being copied from various files. The following shows how this program behaves when it is run:

```
@DDT  
:  
T BMAKE-ICL.  
IDENT. 02X 622  
START;G  
  
RUNNING TABLE: RT.  
CONTEXT BLOCK: CB.  
MIB: MIB.  
MONITOR: MONITOR.  
LISTENER: LISTENER.  
APU CODE: APU.  
NEW ICL FILE: "NEW-ICL".  
  
DONE.  
ZF
```

The various files named in the preceding transaction which are the constituents of the ICL are themselves made in different ways. We consider first the MONITOR files. The MONITOR is compiled with a special version of SPL called here "MSPL". The MONITOR source exists in 9 separate packages which are read in and compiled into a dump file. The dump file is then separated into several of the

ICL-constituent files by means of BXFER, which exists as a QSPL binary file and must be loaded and run by QRUN.

@MSPL

SPL VERSION 1.1 (7/5/73)

```
:.B 610000B
610 MS 16 PAGE FAULTS
:.S E G
0 MS 0 PAGE FAULTS
:R COMM
1569 LINES = 0 WORDS
:R SPEL
247 LINES = 0 WORDS
:R TRAP
158 LINES = 0 WORDS
:R PMS
1309 LINES = 0 WORDS
:R IWS
649 LINES = 0 WORDS
:R FLS
1433 LINES = 0 WORDS
:R CHIO
247 LINES = 0 WORDS
:R SPS
747 LINES = 0 WORDS
:R STRT
330 LINES = 0 WORDS
:.C
943100 MS 11001 PAGE FAULTS → L U
:.P
150584-18224=132360 WORDS
19701 MS 497 PAGE FAULTS
:.D "DMONITOR"
84 PAGES
10985 MS 93 PAGE FAULTS
:.L M
MONITOR RING
WGS 10 OF 600160-602777 RSGS 3010 OF
CS 610000-652331
250 MS 1 PAGE FAULTS
:.F
@QRUN
:T BXFER.
IDENT. 06X 1713
.GET:G
```

*It is important on a new compilation to check to see if the end of SAVE'SP STATE (last fn in regular monitor) has fallen off the page boundary. If so, then recompilation w/ a smaller E factor is req'd.*

L U  
*(should have only runtime failure & long compare)*

DUMP FILE: DMONITOR.  
DATA FILE: "CB".

VIRTUAL ADDRESS: 600000B  
NUMBER OF PAGES: 1.

OK

.GET:G

DUMP FILE: DMONITOR.  
DATA FILE: "MONITOR".

VIRTUAL ADDRESS: 604000B  
NUMBER OF PAGES: 10.

OK

.GET:G

DUMP FILE: DMONITOR.  
DATA FILE: "MIB".

VIRTUAL ADDRESS: 710000B  
NUMBER OF PAGES: 1.

OK

%F

*SPL 827-50* It is necessary to recompile part of the MONITOR with old SPL so that the resident tables can be initialized by running the function INIT'CHT. Old SPL is a 940 dump file. Before it can be RECOVERed, it is necessary to link DDT:PCMD to #2:EMULATOR:PIF and DDT:9SUB to #2:DDT:9SUB. Then:

@RECOVER OLD-SPL JACK

```
40:B
BRS 146D;U
$
M1MODE / 1 0
%!
.SPL:G
```

SPL 8-27.50

ok now reason  
 For some obscure reason  
 address in the trap entry  
 point, 604002B, is occasionally  
 wrong. It is usually one word too  
 big. Fix it if necessary.

```
:.A 2
250 MS
:.B 610000B
2616 MS
:R COMM
1591 LINES = 18608 WORDS
:R SPCL
247 LINES = 1884 WORDS
:R STRT
323 LINES = 3084 WORDS
:.C

CHTD COMPILEING
0 ERRORS 6ROW2836M 0 WGS 13 RSGS
.
.
.
:- INIT'CHT()
COMPILE 1-GO

EXIT TASK 1
-:.D "XMON"
37 PAGES
6266 MS
:.F

%F

@QRUN

:T BXFER.
IDENT. 06X 1713
.GET;G

DUMP FILE: XMON.
DATA FILE: "RT".

VIRTUAL ADDRESS: 664000B
NUMBER OF PAGES: 1.

OK

%F
```

This completes the MONITOR-related files.

The LISTENER is made using yet another version of SPL for compiling code in the Utility ring. This version is called here "USPL". The following shows how it is done:

@USPL

SPL VERSION 1.1 (7/5/73)

:.B 435000B  
610 MS 16 PAGE FAULTS  
:.S E 5  
0 MS 0 PAGE FAULTS  
:.Z S  
0 MS 0 PAGE FAULTS  
:R LISTENER  
  
1815 LINES = 0 WORDS  
.C  
91459 MS 1117 PAGE FAULTS  
.P  
43386-8177=35209 WORDS  
5290 MS 131 PAGE FAULTS  
.D "DLISTENER"  
28 PAGES  
3652 MS 34 PAGE FAULTS  
.L M  
UTILITY RING  
WGS 1726 OF 403200-406777 RSGS 711 OF  
CS 435000-446034  
0 MS 1 PAGE FAULTS  
:-0\$403012B  
2435520B  
-:::F  
@QRUN  
;T BXFER.  
IDENT. 06X 1713

.GET:G

DUMP FILE: DLISTENER.  
DATA FILE: "LISTENER".

VIRTUAL ADDRESS: 434000B  
NUMBER OF PAGES: 3.

OK

%F

It is necessary to modify the sub-process entry-point contents of the CB file made previously to be compatible with the LISTENER just compiled. That is why the contents of 403012B was just examined. This is done as follows:

@DDT

```
CLEAR;U
$
BRS 15D;U
CB.%%
10\ 3
4000;A
;A;X
BIO 10;U
%%
BRS 17D;U
$
7012B[ 2435620 2435520
CLEAR;U
$
BRS 16D;U
CB.%%
4000;A
;A;X
BIO 10;U
%%
BRS 17D;U
$
%F
```

Finally, it is necessary to make the APU code file. A special version of NARP called APUAS is used to assemble the source language. APUAS is a 940 dump file. The source file actually contains two program packages which must be separately assembled. The following script shows the entire production process:

```
@QED

*READ FROM AMCO4.
18766 WORDS.
*:UMCCD:-1WRITE ON "AMCCD".
11147 WORDS.
*:UMCCD:,$WRITE ON "UMCCD".
7620 WORDS.
*FINISHED.
```

```
@REC APUAS
```

```
SOURCE FILE: AMCCD.
OBJECT FILE: "B1".
```

```
AMCCD IDENT
AMCCD COMPILING
HALF WAY
```

50 SEC O ERR 2457 (1327) WRD (S:670,0:0,  
M:1977,U:0)

@CONT  
#2:NARP

SOURCE FILE: UMCCD.  
OBJECT FILE: "B2".

UMCCD IDENT

29 SEC O ERR 2513 (1355) WRD (S:697,0:0,  
M:1977,U:35)

@DDT

O;T B1.  
IDENT. MACCD 2457

;T B2.  
IDENT. UMCCD 5172

;U

%R 23250000 0  
2325,0;R  
240;F  
CLEAR;U  
\$  
BRS 16D;U  
"APU:9BIN".\$\$  
10\ 3  
10000;A  
:A:X  
BIO 10;U  
\$\$  
BRS 17D;U  
\$  
%F