z/OS Control Blocks and the Rexx Storage BIF

René Vincent Jansen, 35th International Rexx Language Symposium 2024, Brisbane, Queensland, Australia
1 Why is this relevant
Some very good reasons to look into this

2 What is a Control Block
The data part of an Operating System

3 What can we do with them
Explore, gain knowledge, understand and diagnose problems, build useful tools

4 View Control Blocks: ISRDDN
ISRDDN suprises again with useful functionality

5 View Control Blocks: IPCS
The old standby for OS analysis and dump formatting

6 Program with the Rexx
STORAGE built-in function
Make you own useful and to-the-point programs

7 Macro's and Assembler
Make you own useful and to-the-point programs, and here the mapping is done for you!

8 .. or even in COBOL
Among other, how to find out if you are running under CICS or JES2
Why is this relevant
Relevant because

- Most performance monitor software reads these
- Can zoom in for specific investigations
- Can roll your own performance tool
- Know how the ASCB tool works

Learning: by looking into the structure of the OS you will understand performance issues better
No 2

What are Control Blocks
What is an operating system

- A Supervisor
- A Scheduler
- Utilities, loaders, linkers and compilers and other small fry

- The control blocks are the data areas (variables) of the supervisor and the scheduler

- Like JCL is the way to command the scheduler
What is Virtual Storage

- Illusion arranged by hardware and system software
- Every address space is 16MB (24bit), 2GB (31bit) or 18 ExaBytes (18 Quintillion bytes (64bit))
- A map divided in different areas, some do overlap
- z/OS has private and common areas
- Some common areas map to the same real storage
- (Different virtual addresses can even map to the same real address)
Virtual Memory Addresses of Pages common to two different users need not be the same. They share two same page in core storage (for example, page X of User 1 and page Y of User 2 resolve to page Z in core storage).

User Terminal Number 1
User Terminal Number 2
User Terminal Number 3
User Terminal Number 4

User is charged only for CPU usage time not for wait time.

CPU 1
CPU 2
CPU 3
CPU 4

Logical Prog A
A’s Map

Logical Prog B
B’s Map

Common Page Pooler

0
1
2

Figure 4. Example of mixed code

Virtual Memory A

Virtual Memory B

Physical Prog. A

Physical Prog. B

Physical Intersection of Virtual Memory

Working Storage Pages are on Drum
Resident

Original Pages are on Auxiliary Files

Supervisor

Disk Packs

Overflow Paging

Needed Pages are already in Core

Recovery Paging

Overflow Paging

Recovery Paging

(Example of Capacity 750-190 Pages where one page 4096 Bytes)
31-bit memory map (MVS/XA)
* write-protected
How does z/OS find programs?

When a program is requested through a system service (like LINK, LOAD, XCTL, or ATTACH) using default options, the system searches for it in the following sequence:

1. **Job pack area (JPA)** A program in JPA has already been loaded in the requesting address space. If the copy in JPA can be used, it will be used. Otherwise, the system either searches for a new copy or defers the request until the copy in JPA becomes available. (For example, the system defers a request until a previous caller is finished before reusing a serially-reusable module that is already in JPA.)

2. **TASKLIB** A program can allocate one or more data sets to a TASKLIB concatenation. Data sets concatenated to TASKLIB are searched for after JPA but before any specified STEPLIB or JOBLIB. Modules loaded by unauthorized tasks that are found in TASKLIB must be brought into private area virtual storage before they can run. Modules that have previously been loaded in common area virtual storage (LPA modules or those loaded by an authorized program into CSA) must be loaded into common area virtual storage before they can run. For more information about TASKLIB, see z/OS MVS Programming: Assembler Services Guide.

3. **STEPLIB or JOBLIB** STEPLIB and JOBLIB are specific DD names that can be used to allocate data sets to be searched ahead of the default system search order for programs. Data sets can be allocated to both the STEPLIB and JOBLIB concatenations in JCL or by a program using dynamic allocation. However, only one or the other will be searched for modules. If both STEPLIB and JOBLIB are allocated for a particular jobstep, the system searches STEPLIB and ignores JOBLIB. Any data sets concatenated to STEPLIB or JOBLIB will be searched after any TASKLIB but before LPA. Modules found in STEPLIB or JOBLIB must be brought into private area virtual storage before they can run. Modules that have previously been loaded in common area virtual storage (LPA modules or those loaded by an authorized program into CSA) must be loaded into common area virtual storage before they can run. For more information about JOBLIB and STEPLIB, see z/OS MVS JCL Reference.

4. **LPA**, which is searched in this order:
   - Dynamic LPA modules, as specified in PROGxx members
   - Fixed LPA (FLPA) modules, as specified in IEAFIXxx members
   - Modified LPA (MLPA) modules, as specified in IEALPAXxx members
   - Pageable LPA (PLPA) modules, loaded from libraries specified in LPALSTxx or PROGxx

   LPA modules are loaded in common storage, shared by all address spaces in the system. Because these modules are reentrant and are not self-modifying, each can be used by any number of tasks in any number of address spaces at the same time. Modules found in LPA do not need to be brought into virtual storage, because they are already in virtual storage.

5. **Libraries in the linklist**, as specified in PROGxx and LNKLSTxx. By default, the linklist begins with SYS1.LINKLIB, SYS1.MIGLIB, SYS1.CSSLIB, SYS1.SIEALNKE, and SYS1.SIEAMIGE. However, you can change this order using SYSLIB in PROGxx and add other libraries to the linklist concatenation. The system must bring modules found in the linklist into private area virtual storage before the programs can run.
Address Space Control Block — ASCB

The ASVT contains an entry for each potential address space. Each entry points to an ASCB, which contains job-related data. The following fields in the ASCB are of interest:

- **ASCBSEQN**: The sequence number of this ASCB on the dispatching queue. Valid only if the address space is currently swapped-in.
- **ASCBDP**: The current dispatching priority for this address space. Valid only if the address space is swapped-in.
- **ASCBEJST**: This doubleword (in time-of-day clock format) represents the total task time received by this address space.
- **ASCBSWCT**: Contains a count of the number of short waits issued by this address space. This value is used in the APG mean-time-to-wait calculation.
- **ASCBVSC**: Contains a count of the total number of VIO slots allocated within the page data sets for this address space.
- **ASCBNVSC**: Contains a count of the total number of non-VIO slots allocated within the page data sets to this address space.
- **ASCBFMCT**: Contains a count of the number of real storage page frames currently occupied by this address space.
- **ASCBJBNI**: Contains a pointer to the 8-character jobname for a batch job. Zero if not a batch job.
- **ASCBJBNS**: Contains a pointer to the 8-character jobname for started tasks, mounts, and TSO users.
- **ASCBSRBT**: This doubleword (in time-of-day clock format) contains the SRB time accumulated by this address space.
Table 1. Structure ASCB

<table>
<thead>
<tr>
<th>Offset</th>
<th>Offset</th>
<th>Type</th>
<th>Len</th>
<th>Name(Din)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>STRUCTURE</td>
<td>0</td>
<td>ASCB</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>DBL WORD</td>
<td>8</td>
<td>ASCBBEGIN(0)</td>
<td>- BEGINNING OF ASCB</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>CHARACTER</td>
<td>4</td>
<td>ASCBASCB</td>
<td>- ACRONYM IN EBCDIC - ASCB-</td>
</tr>
<tr>
<td>4</td>
<td>(4)</td>
<td>ADDRESS</td>
<td>4</td>
<td>ASCBFWD</td>
<td>- ADDRESS OF NEXT ASCB ON ASCB READY QUEUE</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>ADDRESS</td>
<td>4</td>
<td>ASCBBWD</td>
<td>- ADDRESS OF PREVIOUS ASCB ON ASCB READY QUEUE</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td>ADDRESS</td>
<td>4</td>
<td>ASCBLTCS</td>
<td>- TCB and preemptable-class SRB local lock suspend service queue. Serialization: ASCB CML promotion WEB lock.</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>DBL WORD</td>
<td>8</td>
<td>ASCBRO18(0)</td>
<td>Reserved as of z/OS 1.12</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>DBL WORD</td>
<td>8</td>
<td>ASCBSUPC_PREZOS12(0)</td>
<td>- SUPERVISOR CELL FIELD</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>ADDRESS</td>
<td>4</td>
<td>ASCBSVRB_PREZOS12</td>
<td>- SVRB POOL ADDRESS.</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>SIGNED</td>
<td>4</td>
<td>ASCBSYNC_PREZOS12</td>
<td>- COUNT USED TO SYNCHRONIZE SVRB POOL.</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>ADDRESS</td>
<td>4</td>
<td>ASCBIOSP</td>
<td>- POINTER TO IOS PURGE INTERFACE CONTROL BLOCK (IPICB) (MD308)</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>BITSTRING</td>
<td>4</td>
<td>ASCBQWK(0)</td>
<td>WEB QUEUE LOCK WORD</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>BITSTRING</td>
<td>2</td>
<td>ASCBRO1C</td>
<td>RESERVED, MUST BE ZERO</td>
</tr>
</tbody>
</table>

(But I think the PDF books are preferable)

Chapter 1. MVS Data Areas (AIX - IAO) 543
What can we do with them
Activities

- Follow chains from anchors
- Format fields
- Extract real-time information
- Correlate values with events
- Draw conclusions about resource usage and serialization delays

- When using SDSF and RMF(II, III), you look into pre-cooked views of control blocks
  - And more challenging endeavours, to be shown hereafter
### RMF II Address Space Resource Data

<table>
<thead>
<tr>
<th>JOBNAME</th>
<th>CONN</th>
<th>16M</th>
<th>2G</th>
<th>FF</th>
<th>CSF</th>
<th>M</th>
<th>R</th>
<th>ABS</th>
<th>TIME</th>
<th>CPU</th>
<th>EXCP</th>
<th>SWAP</th>
<th>LPA</th>
<th>CSA</th>
<th>NVI</th>
<th>VGM</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>MASTER</em></td>
<td>0.000</td>
<td>0.000</td>
<td>201</td>
<td>628</td>
<td>154</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCAUTH</td>
<td>0.000</td>
<td>0.000</td>
<td>44</td>
<td>4</td>
<td>79X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RASP</td>
<td>0.000</td>
<td>0.000</td>
<td>16</td>
<td>326</td>
<td>53X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRACE</td>
<td>0.000</td>
<td>0.000</td>
<td>29</td>
<td>1037</td>
<td>68X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUMPSRV</td>
<td>0.000</td>
<td>0.000</td>
<td>42</td>
<td>8</td>
<td>156X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCFAS</td>
<td>0.000</td>
<td>0.000</td>
<td>100</td>
<td>423</td>
<td>2038X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRS</td>
<td>0.000</td>
<td>0.000</td>
<td>34</td>
<td>65</td>
<td>149X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMSPDSE</td>
<td>0.000</td>
<td>0.000</td>
<td>46</td>
<td>115</td>
<td>257X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSOLE</td>
<td>0.000</td>
<td>0.000</td>
<td>15</td>
<td>86</td>
<td>114X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WLM</td>
<td>0.000</td>
<td>0.000</td>
<td>75</td>
<td>56</td>
<td>213X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANTMN</td>
<td>0.000</td>
<td>0.000</td>
<td>29</td>
<td>6</td>
<td>214X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANTAS000</td>
<td>0.000</td>
<td>0.000</td>
<td>31</td>
<td>6</td>
<td>184X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEVMAN</td>
<td>0.000</td>
<td>0.000</td>
<td>19</td>
<td>8</td>
<td>69X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMVS</td>
<td>0.000</td>
<td>0.000</td>
<td>111</td>
<td>171</td>
<td>279X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JESXCF</td>
<td>0.000</td>
<td>0.000</td>
<td>24</td>
<td>10</td>
<td>101X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALLOCAS</td>
<td>0.000</td>
<td>0.000</td>
<td>3</td>
<td>4</td>
<td>121X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>0.000</td>
<td>0.000</td>
<td>22</td>
<td>4</td>
<td>93X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODAS</td>
<td>0.000</td>
<td>0.000</td>
<td>75</td>
<td>57</td>
<td>106X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IXLOGR</td>
<td>0.000</td>
<td>0.000</td>
<td>47</td>
<td>18</td>
<td>204X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AKR</td>
<td>0.000</td>
<td>0.000</td>
<td>25</td>
<td>8</td>
<td>109X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CEA</td>
<td>0.000</td>
<td>0.000</td>
<td>23</td>
<td>20</td>
<td>110X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMF</td>
<td>0.000</td>
<td>0.000</td>
<td>25</td>
<td>8</td>
<td>209X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESOLVER</td>
<td>0.000</td>
<td>0.000</td>
<td>25</td>
<td>12</td>
<td>108X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLA</td>
<td>0.000</td>
<td>0.000</td>
<td>41</td>
<td>24</td>
<td>109X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JES2</td>
<td>0.000</td>
<td>0.000</td>
<td>11</td>
<td>281</td>
<td>271</td>
<td>474</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VLF</td>
<td>0.000</td>
<td>0.000</td>
<td>22</td>
<td>79</td>
<td>78X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VTAM</td>
<td>0.000</td>
<td>0.000</td>
<td>38</td>
<td>33</td>
<td>128X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NFSC</td>
<td>0.000</td>
<td>0.000</td>
<td>28</td>
<td>8</td>
<td>236X</td>
<td>0.0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- PF 1=HELP, 2=SPLIT, 3=END, 4=RETURN, 5=RFIND, 6=SORT
- PF 7=UP, 8=DOWN, 9=SWAP, 10=LEFT, 11=RIGHT, 12=RETRIEVE
## RMF II Device Activity

### System: SOM1 Total

<table>
<thead>
<tr>
<th>Command</th>
<th>CPU</th>
<th>9/190 UIC</th>
<th>65K PR</th>
<th>0</th>
<th>System: SOM1 Total</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>11:43:36 I = 5% DEV</strong></td>
<td>ACTV</td>
<td>RESP</td>
<td>IOSQ</td>
<td>DELAY</td>
<td>PEND</td>
</tr>
<tr>
<td>STG</td>
<td>GRP</td>
<td>VOLSER</td>
<td>NUM</td>
<td>PAV</td>
<td>LCU</td>
</tr>
<tr>
<td>FDRES1</td>
<td>0A80</td>
<td>2.890</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDRES2</td>
<td>0A81</td>
<td>1.800</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDSYS1</td>
<td>0A82</td>
<td>1.963</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDBBN2</td>
<td>0A83</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDBBN3</td>
<td>0A84</td>
<td>0.072</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDBLN2</td>
<td>0A85</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDBLN2</td>
<td>0A86</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDGKL1</td>
<td>0A87</td>
<td>0.109</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDDBA1</td>
<td>0A88</td>
<td>0.181</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDDBA2</td>
<td>0A89</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>DBCLASS</td>
<td>FDDBA3</td>
<td>0A8A</td>
<td>0.036</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDDBAR</td>
<td>0A8B</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDDIS1</td>
<td>0A8C</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDDIS2</td>
<td>0A8D</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDDIS3</td>
<td>0A8E</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDDIS4</td>
<td>0A8F</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDIMC1</td>
<td>0A90</td>
<td>0.036</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDMIU1</td>
<td>0A91</td>
<td>0.054</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDMIU2</td>
<td>0A92</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDMIU3</td>
<td>0A93</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDKN1</td>
<td>0A94</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDPAGA</td>
<td>0A95</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDPAGB</td>
<td>0A96</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDPAGC</td>
<td>0A97</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDPAGD</td>
<td>0A98</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDPAGE</td>
<td>0A99</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDPAGF</td>
<td>0A9A</td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
<tr>
<td>FDPAD1</td>
<td>0A9B</td>
<td>0.654</td>
<td>.000</td>
<td>.000</td>
<td>.000</td>
</tr>
</tbody>
</table>

**PF 1=HELP 2=SPLIT 3=END 4=RETURN 5=RFIND 6=SORT**

**PF 7=UP 8=DOWN 9=SWAP 10=LEFT 11=RIGHT 12=RETRIEVE**
**RMF II Memory Activity incl UIC**

<table>
<thead>
<tr>
<th>TIME</th>
<th>AFC</th>
<th>HI SQA</th>
<th>LPA</th>
<th>LPA CSA</th>
<th>L+C</th>
<th>PRI</th>
<th>LSQA</th>
<th>LSQA</th>
<th>CPU</th>
<th>IN</th>
<th>OUT</th>
<th>OUT</th>
<th>OUT</th>
<th>Q</th>
<th>LOG</th>
<th>RQ</th>
<th>WQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:45:54</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>8</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:45:54</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>8</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:45:55</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>8</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:45:55</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>9</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:45:55</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>9</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:45:55</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>9</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:01</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:01</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:01</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:02</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:02</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:02</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:02</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:03</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:03</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:03</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:03</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:04</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:04</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:04</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:04</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:46:05</td>
<td>1.7M</td>
<td>65K</td>
<td>4.5K</td>
<td>5.1K</td>
<td>76</td>
<td>11K</td>
<td>2K</td>
<td>11k</td>
<td>16K</td>
<td>11</td>
<td>62</td>
<td>17</td>
<td>0</td>
<td>17</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PF 1=HELP   2=SPLIT   3=END   4=RETURN   5=RFIND   6=SORT
PF 7=UP   8=DOWN   9=SWAP   10=LEFT   11=RIGHT   12=RETRIEVE
### Speed (Workflow)

<table>
<thead>
<tr>
<th>Name</th>
<th>Users</th>
<th>Active</th>
<th>Speed</th>
<th>Users</th>
<th>Active</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM</td>
<td>79</td>
<td>1</td>
<td>86</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>ALL TSO</td>
<td>1</td>
<td>0</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>ALL STC</td>
<td>73</td>
<td>0</td>
<td>81</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>ALL BATCH</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>ALL ASCH</td>
<td></td>
<td></td>
<td>NOT AVAIL</td>
<td>1</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>ALL OMVS</td>
<td>5</td>
<td>0</td>
<td>NO WORK</td>
<td>1</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>+PROC</td>
<td>47</td>
<td>0</td>
<td>86</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Exceptions

<table>
<thead>
<tr>
<th>Name</th>
<th>Reason</th>
<th>Critical Val.</th>
<th>Possible Cause or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSM</td>
<td>Not Avail</td>
<td>Job HSM is not running.</td>
<td></td>
</tr>
</tbody>
</table>
## RMF III Delay Report

**Command: V1R13 Delay Report**  
**Scroll: CSR**

**Samples:** 100  
**System:** S0W1  
**Date:** 01/17/24  
**Time:** 11:51:40  
**Range:** 100

<table>
<thead>
<tr>
<th>Name</th>
<th>Service</th>
<th>WFL</th>
<th>USG</th>
<th>DLY</th>
<th>IDL</th>
<th>UKN</th>
<th>----</th>
<th>% Delayed</th>
<th>For</th>
<th>----</th>
<th>Primary</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTAM</td>
<td>S SYSTC</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>99</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RSED</td>
<td>S0 STCLM</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>99</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RSE09</td>
<td>0 SRVHIM</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>99</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HTTPD1</td>
<td>S0 STCLM</td>
<td>33</td>
<td>2</td>
<td>0</td>
<td>97</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TN3270</td>
<td>S0 SYSTC</td>
<td>50</td>
<td>3</td>
<td>3</td>
<td>94</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DBAGADM</td>
<td>S0 STCLM</td>
<td>50</td>
<td>1</td>
<td>1</td>
<td>98</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LCKD</td>
<td>S0 STCLM</td>
<td>50</td>
<td>1</td>
<td>1</td>
<td>98</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RSE02</td>
<td>0 SRVHIM</td>
<td>50</td>
<td>1</td>
<td>1</td>
<td>98</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RMFGAT</td>
<td>S0 SYSTC</td>
<td>67</td>
<td>6</td>
<td>3</td>
<td>91</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AB2217#</td>
<td>B BATMD</td>
<td>75</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>97</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AB2217#</td>
<td>B BATMD</td>
<td>75</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>97</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>MASTER</em></td>
<td>S SYSTEM</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RASP</td>
<td>S SYSTEM</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>XCFAS</td>
<td>S SYSTEM</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CONSOLE</td>
<td>S SYSTEM</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>WLM</td>
<td>S SYSTEM</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>OMVS</td>
<td>S SYSTEM</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMF</td>
<td>S SYSTEM</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LLA</td>
<td>S SYSTC</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AB2217#</td>
<td>B BATMD</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>INIT</td>
<td>S SYSTC</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AB2217#</td>
<td>B BATMD</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>INIT</td>
<td>S SYSTC</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AB2217#</td>
<td>B BATMD</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>INIT</td>
<td>S SYSTC</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>JES2</td>
<td>S SYSTC</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>JES2MDN</td>
<td>SYSTEM</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>94</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Key:***
- **PF 1=HELP**
- **PF 2=SPLIT**
- **PF 3=END**
- **PF 4=RETURN**
- **PF 5=RFIN**
- **PF 6=TOGGLE**
- **PF 7=UP**
- **PF 8=DOWN**
- **PF 9=SWAP**
- **PF 10=BREF**
- **PF 11=FREF**
- **PF 12=RETRIEVE**
SDSF DA (ASCB+JES2 Control Blocks)
No 4

View Control Blocks: ISRDDN
<table>
<thead>
<tr>
<th>VOLUME</th>
<th>DISPOSITION</th>
<th>ACT</th>
<th>DDNAME</th>
<th>DATA SET NAME</th>
<th>ACTIONS: B E V M F C I Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDSYS1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>ADFPRINT</td>
<td>AUT330.ADFTABL</td>
<td>--------------</td>
</tr>
<tr>
<td>FDSYS1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>DITPLIB</td>
<td>DIT130.DITPLIB</td>
<td>JES2 Subsystem file</td>
</tr>
<tr>
<td>FDSYS1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>INVCONF</td>
<td>AUT330.INVCONF</td>
<td>--------------</td>
</tr>
<tr>
<td>FDDBAR</td>
<td>NEW, DEL</td>
<td>MOD, DEL</td>
<td>ISPCTL1</td>
<td>SYS24015.T144701.RA000.AB2217.R0100084</td>
<td></td>
</tr>
<tr>
<td>FDSYS1</td>
<td>NEW, DEL</td>
<td>MOD, DEL</td>
<td>ISPCTL2</td>
<td>SYS24015.T144701.RA000.AB2217.R0100085</td>
<td></td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>ISPEXEC</td>
<td>ISP.SIPSPEXEC</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>CSQ710</td>
<td>SYS1.SBPXEXEC</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>ISPLIB</td>
<td>CSQ710.SCSQEXEC</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>ISPLIB</td>
<td>GDDM.SADMMOD</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>FMN121</td>
<td>FMNMOD1</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>CSQ710</td>
<td>CSQAUTH</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>CSQ710</td>
<td>AUT330.SINOMOD1</td>
<td>--------------</td>
</tr>
<tr>
<td>FDSYS1</td>
<td>NEW, DEL</td>
<td>MOD, DEL</td>
<td>ISPLST1</td>
<td>SYS24015.T144701.RA000.AB2217.R0100086</td>
<td></td>
</tr>
<tr>
<td>FDDBAR</td>
<td>NEW, DEL</td>
<td>MOD, DEL</td>
<td>ISPLST2</td>
<td>SYS24015.T144701.RA000.AB2217.R0100087</td>
<td></td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>ISPMLIB</td>
<td>ISP.SIPSPMNU</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>SYS1.DFWMM</td>
<td>SYS1.DFWMLIB</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, Keeper</td>
<td>MOD, DEL</td>
<td>SBSGTMENU</td>
<td>SYS1.HRFMSG</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>SBSGTMENU</td>
<td>SYS1.SBPXEXEC</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>SBSGTMENU</td>
<td>SYS1.SBOMENU</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>SBSGTMENU</td>
<td>SYS1.SBLSMGO</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>CSQ710</td>
<td>CSQSMGSE</td>
<td>--------------</td>
</tr>
<tr>
<td>FDSYS1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>SY1.SEDOEMU</td>
<td>SY1.SEDOEMU</td>
<td>--------------</td>
</tr>
<tr>
<td>FDSYS1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>TCP1P.SEZAMENU</td>
<td>TCP1P.SEZAMENU</td>
<td>--------------</td>
</tr>
<tr>
<td>FDSYS1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>GIM.SGMENU</td>
<td>GIM.SGMENU</td>
<td>--------------</td>
</tr>
<tr>
<td>FDSYS1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>ISF.ISFSTLB</td>
<td>ISF.ISFSTLB</td>
<td>--------------</td>
</tr>
<tr>
<td>FDSYS1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>SY1.SEROMENU</td>
<td>SY1.SEROMENU</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>EQQ.SEDOEMU</td>
<td>EQQ.SEDOEMU</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>FN1140.SFAMSEU</td>
<td>FN1140.SFAMSEU</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>FN1121.SFNMENU</td>
<td>FN1121.SFNMENU</td>
<td>--------------</td>
</tr>
<tr>
<td>FDPD1</td>
<td>SHR, KEEP</td>
<td>MOD, DEL</td>
<td>AUT330.SINGIMSG</td>
<td>AUT330.SINGIMSG</td>
<td>--------------</td>
</tr>
</tbody>
</table>

**TSO ISRDDN or, DDLIST**

The default view shows allocated files (ddnames and datastes)
First, allocate a mapping file to DDNAME ISRDDN

; ISRDDN control block location file.
; ISRDDN BROWSE (OS/390 R5 and later) can use the names to locate storage. For example: B LLT or B JESCT+187
; Locations are (fairly) fixed by architecture
; Some might move around to higher storage regions in modern z/OS

ACEE ASXB+C8? Accessor Environment Element
APHT CVT+C2+C8? APF List
ASCB CVT+C7? Address Space Control Block
ASSB ASCB+180? Address Space Secondary Block
ASVT CVT+C2+C2+200 Address Space Vector Table (after prefix)
ASXB ASCB+6C? Address Space Extension Block
CTC RMC+47? System Resources Manager Control Table
CDE RB+C7? Local Cde List
CTLT TSVT+C4+C4+147 TSO parmib table
CMCT RMC+1187 Channel Measurement Control Table
CMFT RMC+47? Channel Path Measurement Table
CSFB ASCB+C3B? Command Scheduling Control Block
CSV TCV+C4+E4? Contents Supervisor Table
CTV 10.? Communications Vector Table
CVTEX CVT+C1+8B? Communications Vector Table Extension
CVTFIX CVT+C1+8B? Communications Vector Table Prefix
DACA JESCT+787?
DFA CVT+C4+C0? Dfp Id Table
DFVT CVT+C4+C0+C2?
DQE SPS+C8? Vsm Descriptor Queue Element (One Of Zillions)
DSAB JSCB+C1+00+C8? Start of dsab chain
ECTV CVT+C8+C7? Extended Communications Vector Table
EDT DACA+60?
GVT CVT+C1+B0? GRS Vector table
HCCT SSCVT+C1+C7? Hasp Common Storage Communication Table

; KID CVT+42+C7? Cpu Information Iodshid
; ICT RMC+87 Job I/O Management Control Table
; JSCB JSCB+1047 Job Control Table
; JESCT CVT+C1+287? Job Entry Subsystem Communication Table
; JESPXT JESCT+C6+47 Pageable Jesct Extension
; JSCB TCB+B4? Job/Step Control Block
; JSTCB TCB+C7C Job Step Tcb
; LLE TCB+C4+C7? Last Load List Element
; LST CVT+C0+C7? Link List Table
; LLTX CSV+C4? Link List Table Volumes???
; LPAQ CVT+C2+C3? Lpa Cde List
; LPAB STGO+33+C4+347? Lpab Information
; LUV EDT+C1+C7? Cmm Storage Management Control
; ODBC ASCB+90+C7? Resources Manager User Control Block
; OUSB ASCB+90+C7? Resources Manager User Swappable Block
; OUSB ASCB+94+C7? Resources Manager User Extension Block
; PCCA PCCA+C8+C8 Physical Configuration Communication Area
; PICT HECT+5A+C8? Initiator List (Changes Frequently With Jes)
; PSOG CVT+C3+C1? Measurement Facility Control Block
; PSA 0? Prefixed Save Area
; PSTB JSCB+C1+B8+C7? Too Protected Step Control Block
; PVF CVT+C1+C4+C7? Rsm Page Vector Table
; RAB ASCB+C1+B8+C7? Rsm Address Space Block
; RAX ASCB+C1+C6+C7? Rsm Address Space Block Extension
; RB TCB+?
; RCV TCV+C3+C0+C7? System Resource Manager Control Area
; RMCA RMC+1+C7? System Resources Manager Control Table
; RMEC RMC+2+C7? System Resources Monitor Table
; RREM RMC+2+B8 Channel Measurement Control Table
; RCTV CVT+C2+C7? Recovery/Termination Control Table
; RCSB CTGO+C4+C7? Service Call Control Block (Scsb)
; SCC JSCB+C1+B8+C7? Step Control Table
; SCTX CTG+C4+C7? Step Control Table Extension
; SCVT CVT+C0+70? Secondary Cvt
; SHDR CVT+C2+C0+C7?
; SIT+47
; SMCA CVT+C4+C7? Smaf Control Table
; SMCA SMCA+C1+B8+C7? Smca Extension
; SPDA SPMD+C8+C7?
; SPRO SPRO+C8+C7? Same As Scvt
; SSCT JESCT+1+C7? Subsystem Communications Vector Table
; SSCT JESCT+1+C7? Subsystem Communications Vector Table
; SSIB JSCB+C3+C7? Life of Job Subsystem Interface Block
; SSVT SSCVT+C1+C7? Subsystem Vector Table
; SVCT Able CTG+C1+C4+C7? Svctable
; SVDCT SVCT+C8+C8? Svc Table
; SVCRT SVCT+C8+C8? Svc Table
; SVCRT SVCT+C8+C8? Svc Update Recording Table
; SVRKS ASCB+C8+C7? Task Control Block
; TCB TCB+?
; TCB TCB+?
; TASK ASCB+C6+C7? Tcb First Save Area
; TAP ASCB+C6+C7? Tcb First Save Area
; TCB TCB+?
; TCB TCB+?
; TCB TCB+?
; TCB TCB+?
; TCB TCB+?
; UPT PSB+C8+C7? Task Control Block
; VTACVT CVTEX+C4+C7? Task Input/Output Table
<table>
<thead>
<tr>
<th>Goal</th>
<th>Command</th>
<th>Control Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Master Catalog</td>
<td>B CVT+100?+8?+40?+34</td>
<td>CVT-&gt;AMBCS-&gt;ACB-&gt;CAXWA+X'34'</td>
</tr>
<tr>
<td>Show your RACF ACEE</td>
<td>B ACEE</td>
<td></td>
</tr>
</tbody>
</table>

Look for some info (that is hard to get elsewhere)
Also modules; Zoom in on IKJEFT25

- IKJEFT25, the TSO TIME command
- Relevant for performance because it gives you spent service units
  - A service unit is a cpu-independent measure of resource usage
- browse IKJEFT25
- disasm
Fortunately, we have the source of an older version

We can see:
- It is written in PL/S
- The eyecatcher says 76.163
- It is reenterable
- Register equates with @

In SYS1.LINKLIB we see that it has the attributes RF RE RU
View Control Blocks: IPCS
IPCS is the built-on dump analyzer of z/OS; it can also regard active memory as a dump dataset and format control blocks - and lots of other things, like running chains and catching ECB (POST/WAIT) problems with thread (TCB) locking. It has a great relevance for debugging this type of performance problem. On the other hand, nobody knows how to use it anymore and it is relegated to being a tool for the IBM CE; with sites that are read-protecting SYS1.PARMLIB you are out of luck because it needs to read its configuration from there. Already present in the first releases of MVS, and before those
Make Rexx programs with the STORAGE built-in function
STORAGE returns length bytes of data from the specified address in storage. The address is a character string containing the hexadecimal representation of the storage address from which data is retrieved.

The address can be a 31-bit address, represented by 1 to 8 hexadecimal characters. The address can also be a 64-bit address represented by 9 to 17 characters which consists of 8 to 16 hexadecimal characters plus an optional underscore ("_") character separating the high order half and low order half of the 64-bit address. If an "_" is part of the 64-bit address, it must be followed by exactly 8 hexadecimal digits in the low order (or right) half of the 64-bit address.

Optionally, you can specify length, which is the decimal number of bytes to be retrieved from address. The default length is one byte. When length is 0, STORAGE returns a null character string.

If you specify data, STORAGE returns the information from address and then overwrites the storage starting at address with data you specified on the function call. The data is the character string to be stored at address. The length argument has no effect on how much storage is overwritten; the entire data is written.

If the REXX environment under which STORAGE is executing is configured to allow STORAGE to run in read-only mode, then the STORAGE function can be used to read but not alter storage. In this case, do not specify a data argument. If you do specify a new value in the third argument while executing in read-only mode, error message IRX0241I will be issued and the STORAGE function will end in error.

You can use the STORAGE function in REXX execs that run in any MVS address space (TSO/E and non-TSO/E).
**Examples:**

The following are some examples of using STORAGE:

1. To retrieve 25 bytes of data from address 000AAE35, use the STORAGE function as follows:

   ```
   storret = STORAGE(000AAE35, 25)
   ```

2. To replace the data at address 0035D41F with ‘TSO/E REXX’, use the following STORAGE function:

   ```
   storrep = STORAGE(0035D41F, , 'TSO/E REXX')
   ```

   This example first returns one byte of information found at address 0035D41F and then replaces the data beginning at address 0035D41F with the characters ‘TSO/E REXX’.

**Note:** Information is retrieved before it is replaced.

3. Some areas may be accessible to be fetched but not written. That storage can be read as the actual hex data. You can then use the X2D function to then display that hex data in displaceable character format.

   ```
   say ’<‘C2X(STORAGE(10,4))’>’ /* Returns <00FDC248>, perhaps. This area in PSA is update protected, but not fetch protected. The CVT addr.*/
   ```

   Trying to update this same area will fail because address x’10’ is a write protected area in PSA at PSA +x’10’.

   ```
   say ’<‘C2X<STORAGE(10,4,’XXXXXXXX’)’>’ /* Returns <> (a null string) because the storage at x’10’ is at PSA+x’10’ and is write protected and cannot be overwriten by STORAGE */
   ```

4. STORAGE can access 31-bit storage (including 24-bit areas), as well as 64-bit storage. The following shows some possible STORAGE addresses, and the resulting binary addresses that is actually accessed by the STORAGE function.
Simple Job Name exec (works on modern z/OS)

```rexx
/* REXX */

ASCB = C2D(STORAGE(224,4))
ASSB = C2D(STORAGE(D2X(ASCB+336),4))
JSAB = C2D(STORAGE(D2X(ASSB+168),4))
JBNM = STORAGE(D2X(JSAB+28),8)
JBID = STORAGE(D2X(JSAB+20),8)
USID = STORAGE(D2X(JSAB+44),8)
SAY "JOBNAME="JBNM" JOBID="JBID" USERID="USID"
```
There is more than one way that leads to Rome - this works on all known releases of MVS, OS/390 and z/OS

This lists all the active TSO users on the system (all address spaces where CSCB+28 contains a 01

```
000001 /* REXX - BY MOSHIX */
000002 O = 0
000003 SAY 'Currently active users:'
000004 SAY '----------------------'
000005 CVT=PTR(16)
000006 ASVT=PTR(CVT+556)+512 /* GET ASVT */
000007 ASVTMAXU=PTR(ASVT+4) /* GET MAX ASVT ENTRIES */
000008 DO A = 0 TO ASVTMAXU - 1
000009 ASCB=STG(ASVT+16+A*4,4) /* GET PTR TO ASCB (SKIP MASTER) */
000010 IF BITAND(ASCB,'80000000'X) = '00000000'X THEN /* IF IN USE */
000011 IF CTRKID='01'X THEN /* IF TSO USER */
000012   DO
000013      ASCB=C2D(ASCB) /* GET ASCB ADDRESS */
000014      CSCB=PTR(ASC+56) /* GET CSCB ADDRESS */
000015      CHTRKID=STG(CSCB+28,1) /* CHECK ADDR SPACE TYPE */
000016      IF CHTRKID='01'X THEN /* IF TSO USER */
000017         DO
000018            ASCJBNS=PTR(ASC+176) /* GET ASCJBNS */
000019            ASCBSRB=PTR(ASC+200) /* GET ASCBEATT */
000020            O = O + 1
000021      SAY RIGHT(0,2,'0') ASCBSRB,
000022      STG(ASCJBNS,8) /* WE IS SOME HAPPY CAMPER! */
000023      END
000024    END
000025   END
000026 EXIT
000027 PTR: RETURN C2D(STORAGE(D2X(ARG(1)),4)) /* RETURN A POINTER */
000028 STG: RETURN STORAGE(D2X(ARG(1)),ARG(2)) /* RETURN STORAGE */
```
You can run that from USS also

It's the same Rexx interpreter, with added functions in the ADDRESS SYSTEM environment.
Macro mappings and Assembler
Assembler, plain - gets the current job number

```
000010 JOBNR CSECT
000011 REQEQ
000012 USING JOBNR,R12
000013 SAVE (14,12)
000014 LR R12,R15
000015 ST R13,SVAREA+4
000016 LA R15,SVAREA
000017 ST R15,8(R13)
000018 LR R13,R15
000019 DISPLAY PGMSTART
000020 *
000021 L R10,540 CURRENT TCB
000022 L R10,180(R10) POINT TO JFCB
000023 L R10,316(R10) POINT TO SSID
000024 MVC JOBNR,12(R10) COPY TO JOBNUMBER
000025 DISPLAY JOBNR
000026 DISPLAY PGMEND
000027 *
000028 L R13,SVAREA+4
000029 RETURN (14,12),T,RC=8
000030 *
000031 PGMSTART DC CL24'PROGRAM JOBNR STARTED'
000032 PGMEND DC CL24'PROGRAM JOBNR ENDED. '
000033 JOBNR DC CL8' '
000034 SVAREA DC 18F'0'
000035 LTORG LITERALS USED
000036 *
000037 END JOBNR
```
Fully automated using an OS macro

```
* 
* EXTRACT TCBINFO,STCB1ADR,FIELDS=(ASID,PRI,CMC)
...
STCB1ADR DS F
TCBINFO  DS  OF
TCBASID  DS  F
TCBPRIO  DS  F
TCBCMCD  DS  F
```
Assembler, using EXTRACT macro

```
000010  EXTR   CSECT
000011    REGEQ
000012    USING  EXTR,R12
000013    SAVE  (14,12)
000014     LR   R12,R15
000015     ST   R13,SVAREA+4
000016     LA   R15,SVAREA
000017     ST   R15,8(R13)
000018     LR   R13,R15
000019    DISPLAY PGMSTART
000020    PRINT GEN
000021    EXTRACT TCBINFO,"S",FIELDS=(ASID,PRI,CMC)
000022    PRINT NOGEN
000023    DISPLAY TCBINFO,4,F
000024    DISPLAY TCBASID,4,F
000025    DISPLAY TCBPRIQ,4,F
000026    DISPLAY TBCMCD,4,F
000027    DISPLAY PGMEND
000028    *
000029     L   R13,SVAREA+4
000030    RETURN  (14,12),T,RC=8
000031    *
000032   PGMSTART DC  CL24'PROGRAM EXTR STARTED'
000033   PGMEND   DC  CL24'PROGRAM EXTR ENDED. '
000034   TCBINFO DS  OF
000035   TCBSID DS  F
000036   TCBPRIQ DS  F
000037   TBCMCD DS  F
000038   SVAREA  DC  18F'0'
000039  LTORG   LITERALS USED
000041   END    EXTR
```
Assembler, EXTRACT macro expansion (SVC 40)

```
SDFS OUTPUT DISPLAY AB2217A JOB01306 DSID 102 LINE 75 COLUMNS 02-133
COMMAND INPUT ===> _

106 PRINT GEN
107 EXTRACT TCBINFO,'S',FIELDS=(ASID,PRI,CMC)

00014C 0015C 108+ CNOP 0,4 01-EXTRA
00014C 4510 C15C 109+ BAL 1,**16 01-EXTRA
000150 000005F4 110+ DC A(TCBINFO) 01-EXTRA
000154 00000000 111+ DC A(0) 01-EXTRA
000158 0C 112+ DC AL1(12) FIELD BYTE 01-EXTRA
000159 10 113+ DC AL1(16) . FIELD BYTE 2 20021 01-EXTRA
00015A 0000 114+ DC AL2(0) . 20021 01-EXTRA
00015C 0A28 115+ SVC 40 ISSUE EXTRACT SVC 01-EXTRA
116 PRINT NOGEN
```
The super-duper macro version

- The next slide has the best, most stable version
  - It uses IBM provided macros and mapping
  - So the blocks and offsets might change, but the program keeps working
  - There is not a lot of counting or manual mapping involved
    - Lazy is always better
  - This program is exclusively for TSO (or TSO in Batch) due to the use of the TPUT macro for terminal I/O
43

Part 1

000008    START 0
000009    PRINT GEN
000010    SPO00 EQU 0
000011    MYID CSECT
000012    YREGS
000013    STM R14,R12,12(R13)
000014    LR R12,R15
000015    USING MYID,R12
000016    GETMAIN RU,lv=DATALEN,sp=SPO00,loc=BELOW
000017    * THE ADDRESS OF THE OBTAINED STORAGE IS PLACED INTO REGISTER 1.
000018    ST R13,4(,R1)
000019    ST R1,8(,R13)
000020    LR R13,R1
000021    USING SAVEAREA,R13
000022    RUNCHAIN L R3,16
000023    USING CVT,R3
000024    L R3,CVTTCBP
000025    L R3,4(,R3)
000026    DROP R3
000027    USING TCB,R3
000028    L R3,TCBJSCB
000029    DROP R3
000030    USING IEJSCB,R3
000031    L R3,JSCBPSCB
000032    DROP R3
000033  USING TCB,R3
000034     L  R3,TCBJSRCB    POINT TO JSCB. X'B4' OFF CURRENT TCB
000035     DROP R3
000036  USING IEZJSCB,R3
000037     L  R3,JSCBPSRCB    POINT TO PSCB. X'108' OFF THE JSCB
000038     DROP R3
000039  USING PSCB,R3
000040    MVC  MESSAGE(20),MSGLINE    MOVE TEXT TO VARIABLE AREA
000041    MVC  MESSAGE+13(7),PSCBUSR    MOVE MY USERID INTO MESSAGE
000042     DROP R3
000043     TPUT  MESSAGE,L'MESSAGE    PUT THE WHOLE MESSAGE ON THE TUBE
000044  RETURN DS OH
000045     LR  R1,R13    SET UP FOR SAVEAREA FREEMAIN
000046     L  R13,4(,R13)    POINT TO CALLER'S SAVEAREA
000047    FREEMAIN RU,LV=DATALEN,A=(R1),SP=$PO00
000048    LM  R14,R12,12(R13)    RELOAD THE CALLER'S REGISTERS
000049    BR  R14    RETURN TO CALLER
000050  MSGLINE DC CL20'MY USERID IS                CONSTANT PART OF MESSAGE
000051  *
000052 SAVEAREA DSECT
000053     DS  18F    DEFINE MY SAVEAREA - 18 FULLWORDS
000054 MESSAGE DS CL20    VARIABLE MESSAGE AREA
000055     DS OD    ALIGN ON DOUBLEWORD
000056 DATALEN EQU *-SAVEAREA    DEFINE LENGTH OF VARIABLE STORAGE
000057  *
000058 CVT DSECT=YES    CVT MAPPING MACRO
000059    IKJTCB    TCB MAPPING MACRO
000060    IEJJSRCB    JSCB MAPPING MACRO
000061    IKJPSRCB    PSCB MAPPING MACRO
000062     END
№ 8 … or COBOL
IDENTIFICATION DIVISION.

PROGRAM-ID. C0B2JOB.

AUTHOR. GILBERT SAINT-FLOUR.

DATA DIVISION.

WORKING-STORAGE SECTION.

01 RESULTS.

05 JOB-NAME PIC X(8).

05 PROC-STEP PIC X(8).

05 STEP-NAME PIC X(8).

05 PROGRAM-NAME PIC X(8).

05 PROGRAM-NAME2 PIC X(8).

05 JOB-NUMBER PIC X(8).

05 JOB-CLASS PIC X.

05 MSG-CLASS PIC X.

05 PROGRAMMER-NAME PIC X(20).

05 ACCT1 PIC X(32).

05 USER-ID PIC X(8).

05 GROUP-NAME PIC X(8).

05 USER-NAME PIC X(20).

05 BATCH-OR-CICS PIC X(5).

88 Batch Value 'BATCH'.

88 CICS Value 'CICS'.

05 MICRO-SECONDS PIC S9(15) COMP-3.

01 FOUR-BYTES.

05 FULL-WORD PIC S9(8) BINARY.

05 PTR4        REDEFINES FULL-WORD POINTER.

01 EIGHT-BYTES.

05 DOUBLE-WORD PIC S9(18) BINARY.

LINKAGE SECTION.
000053 01 cb1. 05 ptr1 Pointer Occurs 256.
000054 01 cb2. 05 ptr2 Pointer Occurs 256.

000056

Procedure Division.
000057 PSA
000058 SET ADDRESS OF cb1 TO NULL
000059 TCB
000060 SET ADDRESS OF cb1 TO PTR1(136)
000061 MOVE cb1(317:8) TO EIGHT-BYTES
000062 TIOT
000063 COMPUTE MICRO-SECONDS = DOUBLE-WORD / 4096
000064 TIO
000065 SET ADDRESS OF cb2 TO PTR4(4)
000066 MOVE cb2(1:8) TO JOB-NAME
000067 MOVE cb2(9:8) TO PROC-STEP
000068 MOVE cb2(17:8) TO STEP-NAME
000069 JSCB
000070 SET ADDRESS OF cb2 TO PTR1(46)
000071 MOVE cb2(361:8) TO PROGRAM-NAME
000072 SSIB
000073 SET ADDRESS OF cb2 TO PTR2(80)
000074 MOVE cb2(13:8) TO JOB-NUMBER
000075 PRB
000076 SET ADDRESS OF cb2 TO PTR1(1)
000077 JSCB
000078 SET ADDRESS OF cb2 TO PTR4(66)
000079 JCT
000080 MOVE cb2(48:1) TO JOB-CLASS
000081 MOVE cb2(23:1) TO MSG-CLASS
000082 ACT
000083 MOVE ZERO TO FULL-WORD
000084 MOVE cb2(57:3) TO FOUR-BYTES(2:3)
000085 EXT2
000086 SET ADDRESS OF cb2 TO PTR3
000087 MOVE cb2(25:20) TO PROGRAMMER-NAME
000088 EXT2
000089 MOVE ZERO TO FULL-WORD
000090 MOVE cb2(49:1) TO FOUR-BYTES(4:1)
000091 MOVE cb2(50:FULL-WORD) TO ACCT1
000092 CAUF
000093 IF cb2(21:4) = LOW-VALUES THEN
000094 SET BATCH TO TRUE
ELSE
   SET CICS TO TRUE
END-IF
PSA SET ADDRESS OF CB1 TO NULL
ASCB SET ADDRESS OF CB1 TO PTR1(138)
ASXB SET ADDRESS OF CB2 TO PTR1(28)
MOVE CB2(193:8) TO USER-ID
ACEE SET ADDRESS OF CB2 TO PTR2(51)
MOVE CB2(31:8) TO GROUP-NAMES
UNAM SET ADDRESS OF CB1 TO PTR2(26)
MOVE ZERO TO FULL-WORD
MOVE CB1(1:1) TO FOUR-BYTES(4:1)
MOVE CB1(2:FULL-WORD) TO USER-NAME
DISPLAY JOB-NAME
   PROC-STEP
   STEP-NAME
   PROGRAM-NAMES
   PROGRAM-NAMES2
   JOB-NUMBER
   JOB-CLASS
   MSG-CLASS
   MICRO-SECONDS
DISPLAY QUOTE PROGRAMMER-NAMES QUOTE
QUOTE ACCT1 QUOTE
QUOTE BATCH-OR-CICS
QUOTE USER-ID
QUOTE GROUP-NAMES
QUOTE USER-NAME QUOTE
GOBACK.
Which, like always, is a lot of source code for one line of output. But that is the charm of COBOL: no documentation needed.
The end.
Q?: rv.jansen@xs4all.nl