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 |
№ 1

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
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), 2TB (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 is charged
only for CPU usage time not for wait time.

CPU 1

User Terminal Number 2

User Terminal Number 3

User Terminal Number 4

Waiting for time slice

CPU 2

CPU 3

CPU 4

Working Storage:
Pages are on Drum

Original Pages:
Are on Auxiliary Files

Resident

Supervisor

Drum

Overflow Paging

Disks Packs

Overflow Paging

Recovery Paging

(Example of Capacity:
750–190 Pages where
one Page = 4096 Bytes)

Logical Prog A

A’s Map

Logical Prog B

B’s Map

Common Page Pointer

Figure 4. Example of mixed code

Virtual Memory A

Physical Prog. A

Physical Intersection of Virtual Memory

Virtual Memory B

Physical Prog. B
### Extended Private
- Extended LSQA
- Extended SAA
- Extended 29/230/249
- Extended User Region

### Extended Common
- Extended CSA
- Extended MLPA
- Extended FLPA
- Extended PLPA
- Extended SOA
- Extended NUC
- NUC
- SQA
- PLPA
- MLA
- MLPA
- CSA
- LSQA
- SAA
- 29/230/249 (Authorized User Key)

### Common
- System Region
- PSA

* 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 (PROGxx) modules, as specified in PROGxx members
   - Fixed LPA (FLPA) modules, as specified in IEAFIXx members
   - Modified LPA (MLPA) modules, as specified in IEALPXxx 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.

- **ASCBBDP**  
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.

- **SCBSWCT**  
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.

- **ASCBIN**  
Contains a pointer to the 8-character jobname for a batch job. Zero if not a batch job.

- **ASCBJNS**  
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.
Current day version (nearly unchanged) at https://www.ibm.com/docs/en/zos/2.2.0?topic=information-ascb-mapping

ASCB mapping

Last Updated: 2021-03-22

Table 1. Structure ASCB

<table>
<thead>
<tr>
<th>Offset</th>
<th>Offset</th>
<th>Len</th>
<th>Name(Din)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(B)</td>
<td>0</td>
<td>STRUCTURE</td>
<td>0 ASCB</td>
</tr>
<tr>
<td>0</td>
<td>(B)</td>
<td>8</td>
<td>ASCB8BEGIN(0)</td>
<td>BEGINNING OF ASCB</td>
</tr>
<tr>
<td>0</td>
<td>(B)</td>
<td>4</td>
<td>CHARACTER</td>
<td>4 ASCBASCB</td>
</tr>
<tr>
<td>4</td>
<td>(4)</td>
<td>4</td>
<td>ADDRESS</td>
<td>4 ASCBFdWP</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>4</td>
<td>ADDRESS</td>
<td>4 ASCB8dWP</td>
</tr>
<tr>
<td>12</td>
<td>(C)</td>
<td>4</td>
<td>ADDRESS</td>
<td>4 ASCBLtcs</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>8</td>
<td>ASCBR018(0)</td>
<td>Reserved as of z/OS 1.12</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>8</td>
<td>ASCBSUPC_PREZOS12(0)</td>
<td>SUPERVISOR CELL FIELD</td>
</tr>
<tr>
<td>16</td>
<td>(10)</td>
<td>4</td>
<td>ADDRESS</td>
<td>4 ASCBSvrb_PREZOS12</td>
</tr>
<tr>
<td>20</td>
<td>(14)</td>
<td>4</td>
<td>ADDRESS</td>
<td>4 ASCBSync_PREZOS12</td>
</tr>
<tr>
<td>24</td>
<td>(18)</td>
<td>4</td>
<td>ADDRESS</td>
<td>4 ASCBI0sp</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>4</td>
<td>BITSTRING</td>
<td>4 ASCBqWlk(0)</td>
</tr>
<tr>
<td>28</td>
<td>(1C)</td>
<td>2</td>
<td>BITSTRING</td>
<td>2 ASCBR01C</td>
</tr>
</tbody>
</table>

(But I think the PDF books are preferable)

Chapter 1. MVS Data Areas (AWE - IAR) 143
№ 3 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
<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>V&amp;N</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>MASTER</em></td>
<td>0.000</td>
<td>201</td>
<td>628</td>
<td>154</td>
<td>0.000</td>
<td>2615</td>
<td>4480</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>PCAUTH</td>
<td>0.000</td>
<td>44</td>
<td>4</td>
<td>79</td>
<td>X</td>
<td>0.000</td>
<td>0.06</td>
<td>0.07</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>RASP</td>
<td>0.000</td>
<td>16</td>
<td>326</td>
<td>53</td>
<td>X</td>
<td>0.000</td>
<td>0.04</td>
<td>347.9</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>TRACE</td>
<td>0.000</td>
<td>29</td>
<td>1037</td>
<td>68</td>
<td>X</td>
<td>0.000</td>
<td>0.09</td>
<td>0.12</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>DUMPSRV</td>
<td>0.000</td>
<td>42</td>
<td>8</td>
<td>156</td>
<td></td>
<td>0.000</td>
<td>37.83</td>
<td>56.10</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>XCFAS</td>
<td>0.000</td>
<td>100</td>
<td>423</td>
<td>2038</td>
<td>X</td>
<td>0.000</td>
<td>3788</td>
<td>4144</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>GRS</td>
<td>0.000</td>
<td>34</td>
<td>65</td>
<td>149</td>
<td>X</td>
<td>0.000</td>
<td>1.41</td>
<td>47.88</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>SMSPDSE</td>
<td>0.000</td>
<td>46</td>
<td>115</td>
<td>257</td>
<td>X</td>
<td>0.000</td>
<td>650.3</td>
<td>715.7</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CONSOLE</td>
<td>0.000</td>
<td>15</td>
<td>86</td>
<td>114</td>
<td>X</td>
<td>0.000</td>
<td>357.5</td>
<td>430.4</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>WLM</td>
<td>0.000</td>
<td>75</td>
<td>56</td>
<td>213</td>
<td>X</td>
<td>0.000</td>
<td>17503</td>
<td>20330</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>ANTMAIN</td>
<td>0.000</td>
<td>29</td>
<td>6</td>
<td>214</td>
<td>X</td>
<td>0.000</td>
<td>118.4</td>
<td>133.4</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>ANTAS0000</td>
<td>0.000</td>
<td>31</td>
<td>6</td>
<td>184</td>
<td>X</td>
<td>0.000</td>
<td>4.97</td>
<td>5.71</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>DEVMAN</td>
<td>0.000</td>
<td>19</td>
<td>8</td>
<td>69</td>
<td>X</td>
<td>0.000</td>
<td>9.85</td>
<td>15.27</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>OMVS</td>
<td>0.000</td>
<td>111</td>
<td>171</td>
<td>279</td>
<td>X</td>
<td>0.000</td>
<td>2466</td>
<td>2572</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>JESXC</td>
<td>0.000</td>
<td>24</td>
<td>10</td>
<td>101</td>
<td>X</td>
<td>0.000</td>
<td>322.8</td>
<td>458.5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>ALLOCAS</td>
<td>0.000</td>
<td>3</td>
<td>4</td>
<td>121</td>
<td>X</td>
<td>0.000</td>
<td>0.81</td>
<td>0.82</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>0.000</td>
<td>22</td>
<td>4</td>
<td>93</td>
<td>X</td>
<td>0.000</td>
<td>1685</td>
<td>1708</td>
<td>9.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>IDASAS</td>
<td>0.000</td>
<td>75</td>
<td>57</td>
<td>106</td>
<td>X</td>
<td>0.000</td>
<td>389.1</td>
<td>461.2</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>IXGLOGR</td>
<td>0.000</td>
<td>47</td>
<td>18</td>
<td>204</td>
<td>X</td>
<td>0.000</td>
<td>574.6</td>
<td>633.6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>AKR</td>
<td>0.000</td>
<td>25</td>
<td>8</td>
<td>109</td>
<td>X</td>
<td>0.000</td>
<td>1.58</td>
<td>1.79</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>CEA</td>
<td>0.000</td>
<td>23</td>
<td>20</td>
<td>110</td>
<td>X</td>
<td>0.000</td>
<td>4.76</td>
<td>5.30</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>SMF</td>
<td>0.000</td>
<td>25</td>
<td>8</td>
<td>209</td>
<td>X</td>
<td>0.000</td>
<td>11.56</td>
<td>367.6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>RESOLVER</td>
<td>0.000</td>
<td>25</td>
<td>12</td>
<td>108</td>
<td>X</td>
<td>0.000</td>
<td>10.07</td>
<td>13.66</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>LLA</td>
<td>0.000</td>
<td>41</td>
<td>24</td>
<td>109</td>
<td>X</td>
<td>0.000</td>
<td>54.46</td>
<td>56.23</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>JES2</td>
<td>0.000</td>
<td>11</td>
<td>281</td>
<td>271</td>
<td>474</td>
<td>0.000</td>
<td>3266</td>
<td>3534</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>VLF</td>
<td>0.000</td>
<td>22</td>
<td>79</td>
<td>78</td>
<td>X</td>
<td>0.000</td>
<td>74.72</td>
<td>86.01</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>TFM</td>
<td>0.000</td>
<td>38</td>
<td>33</td>
<td>128</td>
<td>X</td>
<td>0.000</td>
<td>363.2</td>
<td>481.5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>NFSC</td>
<td>0.000</td>
<td>28</td>
<td>8</td>
<td>236</td>
<td>X</td>
<td>0.000</td>
<td>99.83</td>
<td>117.2</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>
### RMF II Device Activity

**Command ==> _**

**CPU** 9/190  **UIC** 65K  **PR** 0  **System: SOM1 Total**

<table>
<thead>
<tr>
<th>STG GRP</th>
<th>VOLSER NUM</th>
<th>PAV</th>
<th>LCU</th>
<th>ACTV</th>
<th>RESP</th>
<th>IOSQ</th>
<th>-DELAY-</th>
<th>PEND</th>
<th>DISC</th>
<th>CONN</th>
<th>%D</th>
<th>%D</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDRES1</td>
<td>0A80</td>
<td></td>
<td></td>
<td>2.890</td>
<td>.000</td>
<td>.000</td>
<td>.000*.000*</td>
<td>.000*</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDRES2</td>
<td>0A81</td>
<td></td>
<td></td>
<td>1.800</td>
<td>.000</td>
<td>.000</td>
<td>.000*.000*</td>
<td>.000*</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDSYS1</td>
<td>0A82</td>
<td></td>
<td></td>
<td>1.963</td>
<td>.000</td>
<td>.000</td>
<td>.000*.000<em>16</em>16</td>
<td>.000*</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDBBN2</td>
<td>0A83</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDBBN3</td>
<td>0A84</td>
<td></td>
<td></td>
<td>0.072</td>
<td>.000</td>
<td>.000</td>
<td>.000*.000*</td>
<td>.000*</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDBLZ1</td>
<td>0A85</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDBLZ2</td>
<td>0A86</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDC511</td>
<td>0A87</td>
<td></td>
<td></td>
<td>0.109</td>
<td>.000</td>
<td>.000</td>
<td>.000*.000*</td>
<td>.000*</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDDBA1</td>
<td>0A88</td>
<td></td>
<td></td>
<td>0.181</td>
<td>.000</td>
<td>.000</td>
<td>.000*.000*</td>
<td>.000*</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDDBA2</td>
<td>0A89</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DBCLASS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FDDBA3</td>
<td>0A8A</td>
<td></td>
<td></td>
<td>0.036</td>
<td>.000</td>
<td>.000</td>
<td>.000*.000*</td>
<td>.000*</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDDBAR</td>
<td>0A8B</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDDIS1</td>
<td>0A8C</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDDIS2</td>
<td>0A8D</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDDIS3</td>
<td>0A8E</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDDIS4</td>
<td>0A8F</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDIMC1</td>
<td>0A90</td>
<td></td>
<td></td>
<td>0.036</td>
<td>.000</td>
<td>.000</td>
<td>.000*.000*</td>
<td>.000*</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDIMJ1</td>
<td>0A91</td>
<td></td>
<td></td>
<td>0.054</td>
<td>.000</td>
<td>.000</td>
<td>.000*.000*</td>
<td>.000*</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDIMU2</td>
<td>0A92</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDIMU3</td>
<td>0A93</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDKAN1</td>
<td>0A94</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDPAGA</td>
<td>0A95</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDPAGB</td>
<td>0A96</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDPAGC</td>
<td>0A97</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDPAGD</td>
<td>0A98</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDPAGE</td>
<td>0A99</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDPAGF</td>
<td>0A9A</td>
<td></td>
<td></td>
<td>0.000</td>
<td>.000</td>
<td>.000</td>
<td></td>
<td>.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>FDPARD1</td>
<td>0A9B</td>
<td></td>
<td></td>
<td>0.654</td>
<td>.000</td>
<td>.000</td>
<td>.000*.000*</td>
<td>.000*</td>
<td>0*</td>
<td>0</td>
<td>0</td>
<td>0</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**
<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>PQ SQA</th>
<th>PQ SQA CPU</th>
<th>IN</th>
<th>OUT</th>
<th>OUT</th>
<th>OUT</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 16K</td>
<td>8</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>8</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>8</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>9</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>9</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>9</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>10</td>
<td>62</td>
<td>17</td>
<td>0</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 16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</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 16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</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 16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</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 16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</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 16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</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 16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</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 16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</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 16K</td>
<td>10</td>
<td>63</td>
<td>16</td>
<td>0</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 16K</td>
<td>11</td>
<td>62</td>
<td>17</td>
<td>0</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 Active</th>
<th>Speed</th>
<th>NAME</th>
<th>Users Active</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSTEM</td>
<td>79</td>
<td>1</td>
<td>86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALL TSO</td>
<td>1</td>
<td>0</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALL STC</td>
<td>73</td>
<td>0</td>
<td>81</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALL BATCH</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALL ASCH</td>
<td></td>
<td></td>
<td></td>
<td>NOT AVAIL</td>
<td></td>
</tr>
<tr>
<td>ALL OMVS</td>
<td>5</td>
<td>0</td>
<td>NO WORK</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>PROC</em></td>
<td>47</td>
<td>0</td>
<td>86</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></td>
<td>JOB HSM IS NOT RUNNING</td>
</tr>
</tbody>
</table>

**PF Key:**
- 1=HELP
- 2=SPLIT
- 3=END
- 4=RETURN
- 5=RFIN
- 6=TOGGLE
- 7=UP
- 8=DOWN
- 9=SWAP
- 10=BREF
- 11=FREE
- 12=RETRIEVE
<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>% Delayed</th>
<th>FOR</th>
<th>PRIMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTAM</td>
<td>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>
</tr>
<tr>
<td>RSED</td>
<td>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>
</tr>
<tr>
<td>RSED9</td>
<td>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>
</tr>
<tr>
<td>HTTPD1</td>
<td>STCLM</td>
<td>33</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>97</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TN3270</td>
<td>SYSTC</td>
<td>50</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>94</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DBAGADM</td>
<td>STCLM</td>
<td>50</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>98</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lckd</td>
<td>STCLM</td>
<td>50</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>98</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RSED2</td>
<td>SRVHIM</td>
<td>50</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>98</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RMFGAT</td>
<td>SYSTC</td>
<td>67</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>91</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AB2217</td>
<td>BATM</td>
<td>75</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AB2217</td>
<td>BATM</td>
<td>75</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><em>MASTER</em></td>
<td>SYSTEM</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>RASP</td>
<td>SYSTEM</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>XCFAS</td>
<td>SYSTEM</td>
<td>100</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CONSOLE</td>
<td>SYSTEM</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>WLM</td>
<td>SYSTEM</td>
<td>100</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>94</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>QMVS</td>
<td>SYSTEM</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SMF</td>
<td>SYSTEM</td>
<td>100</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>97</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LLA</td>
<td>SYSTC</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AB2217</td>
<td>BATM</td>
<td>100</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>INIT</td>
<td>SYSTC</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AB2217</td>
<td>BATM</td>
<td>100</td>
<td>4</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>SYSTC</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AB2217</td>
<td>BATM</td>
<td>100</td>
<td>4</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>SYSTC</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>48</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>JES2</td>
<td>SYSTC</td>
<td>100</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>JES2MDN</td>
<td>SYSTEM</td>
<td>100</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>94</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

PF 1=HELP  2=SPLIT  3=END  4=RETURN  5=RFIN  6=TOGGLE
PF 7=UP  8=DOWN  9=SWAP  10=BREF  11=FREF  12=RETRIEVE
<table>
<thead>
<tr>
<th>Jobname</th>
<th>Stepname</th>
<th>Task</th>
<th>Owner</th>
<th>Pos</th>
<th>DP</th>
<th>Real</th>
<th>Page</th>
<th>SID</th>
<th>CPU1</th>
<th>ASID1</th>
<th>ASID2</th>
<th>EXCP-Cnt</th>
<th>CPU-Time</th>
<th>SR</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>MASTER</em></td>
<td>STC01078</td>
<td>+MASTER+</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>1860</td>
<td>0.00</td>
<td>0.00</td>
<td>0.26</td>
<td>1</td>
<td>0001</td>
<td>10359</td>
<td>4487.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCAUTH</td>
<td>PCAUTH</td>
<td></td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>138</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2</td>
<td>0002</td>
<td>18</td>
<td>0.07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RASP</td>
<td>RASP</td>
<td></td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>388</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>3</td>
<td>0003</td>
<td>2</td>
<td>348.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRACE</td>
<td>TRACE</td>
<td></td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>1132</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>4</td>
<td>0004</td>
<td>78</td>
<td>0.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUMPSRV</td>
<td>DUMPSRV</td>
<td>DUMPSRV</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>468</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>5</td>
<td>0005</td>
<td>29779</td>
<td>56.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XCFAS</td>
<td>XCFAS</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>3926</td>
<td>0.66</td>
<td>0.26</td>
<td>6</td>
<td>0006</td>
<td>72362</td>
<td>41532</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRS</td>
<td>GRS</td>
<td></td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>1902</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>7</td>
<td>0007</td>
<td>16</td>
<td>47.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMSPS0E</td>
<td>SMSPS0E</td>
<td></td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>5133</td>
<td>0.00</td>
<td>2.08</td>
<td>8</td>
<td>0008</td>
<td>7</td>
<td>716.92</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSOLE</td>
<td>CONSOLE</td>
<td></td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>1856</td>
<td>9.00</td>
<td>3.08</td>
<td>10</td>
<td>000A</td>
<td>631</td>
<td>1835.84</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WLW</td>
<td>WLW</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>1427</td>
<td>0.00</td>
<td>3.08</td>
<td>10</td>
<td>000A</td>
<td>27</td>
<td>20360.07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANTMAIN</td>
<td>ANTMAIN</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>1439</td>
<td>0.00</td>
<td>0.00</td>
<td>11</td>
<td>0008</td>
<td>1553</td>
<td>133.80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANTAS000</td>
<td>ANTAS000</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>CI</td>
<td>1329</td>
<td>0.00</td>
<td>0.00</td>
<td>12</td>
<td>000C</td>
<td>1384</td>
<td>5.71</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEVMAN</td>
<td>DEVMAN</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>423</td>
<td>0.00</td>
<td>0.00</td>
<td>13</td>
<td>000D</td>
<td>595</td>
<td>15.29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMVS</td>
<td>OMVS</td>
<td>DMVS</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>207</td>
<td>0.00</td>
<td>0.07</td>
<td>14</td>
<td>000E</td>
<td>1901</td>
<td>2574.37</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JESXCF</td>
<td>JESXCF</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>635</td>
<td>0.00</td>
<td>0.00</td>
<td>16</td>
<td>0010</td>
<td>685</td>
<td>459.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALLOCAS</td>
<td>ALLOCAS</td>
<td></td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>2490</td>
<td>0.00</td>
<td>0.00</td>
<td>17</td>
<td>0011</td>
<td>9</td>
<td>0.82</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMS</td>
<td>SMS</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>403</td>
<td>0.00</td>
<td>0.00</td>
<td>18</td>
<td>0012</td>
<td>36325</td>
<td>1709.91</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IODAS</td>
<td>IODAS</td>
<td></td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>380</td>
<td>0.00</td>
<td>0.19</td>
<td>19</td>
<td>0013</td>
<td>311</td>
<td>461.83</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INGLGR</td>
<td>INGLGR</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>2741</td>
<td>0.00</td>
<td>0.00</td>
<td>20</td>
<td>0014</td>
<td>1635</td>
<td>634.21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AXR</td>
<td>AXR</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>CI</td>
<td>451</td>
<td>0.00</td>
<td>0.00</td>
<td>21</td>
<td>0015</td>
<td>266</td>
<td>1.79</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CEA</td>
<td>CEA</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>3017</td>
<td>0.00</td>
<td>0.00</td>
<td>22</td>
<td>0016</td>
<td>421</td>
<td>5.30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMF</td>
<td>SMF</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>499</td>
<td>0.00</td>
<td>0.00</td>
<td>23</td>
<td>0017</td>
<td>584</td>
<td>368.04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESOLVER</td>
<td>RESOLVER</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>389</td>
<td>0.00</td>
<td>0.00</td>
<td>24</td>
<td>0018</td>
<td>247</td>
<td>13.67</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LLA</td>
<td>LLA</td>
<td></td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>2636</td>
<td>0.00</td>
<td>0.00</td>
<td>25</td>
<td>0019</td>
<td>10356</td>
<td>58.21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JES2</td>
<td>JES2</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>7978</td>
<td>3.96</td>
<td>1.04</td>
<td>27</td>
<td>001B</td>
<td>53820</td>
<td>3540.46</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VLF</td>
<td>VLF</td>
<td></td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>4267</td>
<td>0.00</td>
<td>0.00</td>
<td>28</td>
<td>001C</td>
<td>154</td>
<td>86.10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VTAM</td>
<td>VTAM</td>
<td>STC01079</td>
<td>START1</td>
<td>NS</td>
<td>FF</td>
<td>2857</td>
<td>0.00</td>
<td>0.07</td>
<td>29</td>
<td>001D</td>
<td>4669</td>
<td>482.27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NFSC</td>
<td>NFSC</td>
<td>MVSCLNT</td>
<td>START2</td>
<td>NS</td>
<td>FF</td>
<td>117</td>
<td>0.00</td>
<td>0.00</td>
<td>30</td>
<td>001E</td>
<td>891</td>
<td>117.30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLF</td>
<td>DLF</td>
<td>DLF</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>290</td>
<td>0.00</td>
<td>0.00</td>
<td>31</td>
<td>001F</td>
<td>391</td>
<td>6.90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RACF</td>
<td>RACF</td>
<td>STC01092</td>
<td>START2</td>
<td>NS</td>
<td>FF</td>
<td>657</td>
<td>0.00</td>
<td>0.00</td>
<td>32</td>
<td>0020</td>
<td>706</td>
<td>80.08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATALOG</td>
<td>CATALOG</td>
<td>IEFPROC</td>
<td></td>
<td>NS</td>
<td>FF</td>
<td>1108</td>
<td>0.00</td>
<td>0.00</td>
<td>33</td>
<td>0021</td>
<td>5819</td>
<td>633.08</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SDSF DA (ASCB+JES2 Control Blocks)**
View Control Blocks: ISRDDN
The default view shows allocated files (ddnames and datastes)
First, allocate a mapping file to DDNAME ISRDDN

ISRDDN control block location file.

If this file is allocated to a dname of ISRDDN the the
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 CSTVT+C28? APF List
ASCB CVT7+C7? Address Space Control Block
ASSB ASXB+C88? Address Space Secondary Block
ASVT CVT7+C28+C200 Address Space Vector Table (after prefix)
ASXB ASXB+C6C? Address Space Extension Block
CCT RMC+47 System Resources Manager Control Table
CDE RB+C7? Local Cde List
CTLT TSVT+C48+C48 TSO parlmt table
CMCT RMC+C1187 Channel Measurement Control Table
CMPT CMPT+C7 Channel Path Measurement Table
CSCE ASCB+C88 Command Scheduling Control Block
CVET EVTM+C48 Contents Supervisor Table
CVT 10.? Communications Vector Table
CVTEX CVT+C188 Communications Vector Table Extention
CVTXFIX CVT+C188 Communications Vector Table Extention Prefix
DACA JESCT+C87?
DFQF CVT+C48D? Dfp Id Table
DFVT CVT+C48+C48 Dfp Table
DSQF SPSVT+C388? Swp Descriptor Queue Element (One Of Zillions)
DSAB JESCT+C48+C48 Start of dsb chain
ECVT CVT+C87 Extended Communications Vector Table
EDT DACA+C80? GRS Vector table
HCCV SSCVT+C1C7 Hapw Common Storage Communication 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

STORAGE ( address , length , data )

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:

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

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

   ```plaintext
   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 displacable character format.

   ```plaintext
   say '<C2X(STORAGE(10,4))>' /* Returns <0FDC248>, 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'.

   ```plaintext
   say '<C2X<STORAGE(10,4,'XXXX'>>)>' /* Returns <> (a null string) because the storage at x'10' is at PSA+x'10' and is write protected and cannot be overwritten 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 */

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

```
000001 /* REXX - BY MOSHIJX */
000002 Q = 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        MASTER)        */
000011 IF BITAND(ASCB,'80000000'X) = '00000000'X THEN /* IF IN USE */
000012        IF TSO USER */
000013        DO
000014        ASCB=C2D(ASCB)        */ GET ASCB ADDRESS */
000015        CSCB=PTR(ASCB+56)        */ GET CSCB ADDRESS */
000016        CHTRKID=STG(CSCB+28,1)    */ CHECK ADDR SPACE TYPE */
000017        IF CHTRKID='01'X THEN    */ IF TSO USER */
000018        DO
000019        ASCJBNS=PTR(ASCB+176)    */ GET ASCJBNS */
000020        ASCBSRBT=PTR(ASCB+200)    */ GET ASCBEATT */
000021        O = O + 1
000022        SAY RIGHT(0,2,'0') ASCBSRBT,
000023        STG(ASCBJBNS,8)    */ WE IS SOME HAPPY CAMPER! */
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
No 7

Macro mappings and Assembler
Assembler, plain - gets the current job number

000010  JOBNBR  CSECT
000011    REGEQ
000012  USING  JOBNBR,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  JOBNBR,12(R10)  COPY TO JOBNUMBER
000025 DISPLAY  JOBNBR
000026  DISPLAY  PGMEND
000027  *
000028    L  R13,SVAREA+4
000029 RETURN  (14,12),T,RC=8
000030  *
000031 PGMSTART  DC  CL24'PROGRAM  JOBNBR  STARTED'
000032 PGMEND  DC  CL24'PROGRAM  JOBNBR  ENDED. '
000033  JOBNR  DC  CL8'
000034  SVAREA  DC  18F'0'
000035  LTORG  LITERALS USED
000036  *
000037   END   JOBNBR
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      REQEQ
000012 USING EXTR,R12
000013 SAVE (14,12)
000014 LR   R12,R15
000015 ST   R13,SVAREA+4
000016 LA   R15,SVAREA
000017 ST   R15,R(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 TCBPRIO,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 TCBASID DS   F
000036 TCBPRIO DS   F
000037 TBCMCD DS   F
000038 SVAREA DC   18F'0'
000039 LTORG LITERALS USED
000041 END   EXTR
Assembler, EXTRACT macro expansion (SVC 40)
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
START 0

WE WANT TO SHOW THE EXPANSIONS

DEFINE SUBPOOL TO BE 0

REGISTER EQUATES

SAVE CALLER’S REGISTERS R14 THRU R12

LOAD ENTRY POINT INTO BASE REGISTER

TELL THE ASSEMBLER, R12 IS THE BASE

GETMAIN RU, LV=DATALEN, SP=SPO00, LOC=BETWEEN

THE ADDRESS OF THE OBTAINED STORAGE IS PLACED INTO REGISTER 1.

SAVE CALLER’S SAVEAREA ADDRESS

STORE OUR SAVEAREA ADDRESS IN HIS

POINT REGISTER 13 TO OUR SAVE AREA

TELL ASSEMBLER

POINT TO CVT. ADDR IS IN LOW STORAGE

POINT TO TCB/ASCB WORDS, "0" OFF CVT

POINT TO TCB, "4" OFF TCB/ASCB WORDS

POINT TO JSCB. X'B4' OFF CURRENT TCB

POINT TO PSCB. X'108' OFF THE JSCB

DROP R3

DROP R3

DROP R3

DROP R3

DROP R3

DROP R3
000033  USING TCB,R3
000034  L   R3,TCBJSCB    POINT TO JSCB. X'B4' OFF CURRENT TCB
000035  DROP R3
000036  USING IEZJSCB,R3
000037  L   R3,JSCBPSCB   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),PSCBUSER  MOVE MY USERID INTO MESSAGE
000042  DROP R3
000043  TPUS 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=SP000
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  IEZJSCB     JSCB MAPPING MACRO
000061  IKJPS CB   PSCB MAPPING MACRO
000062  END
№ 8 … or COBOL
IDENTIFICATION DIVISION.
Program-ID. COB2JOB.
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.
47

MOVE cb1(317:8) TO EIGHT-BYTES

COMPUTE MICRO-SECONDS = DOUBLE-WORD / 4096

SET ADDRESS OF cb1 TO NULL

SET ADDRESS OF cb1 TO PTR1(136)

MOVE cb2(1:8) TO JOB-NAME

MOVE cb2(9:8) TO PROC-STEP

MOVE cb2(17:8) TO STEP-NAME

SET ADDRESS OF cb2 TO PTR1(46)

MOVE cb2(361:8) TO PROGRAM-NAME

SET ADDRESS OF cb2 TO PTR2(80)

MOVE cb2(13:8) TO JOB-NUMBER

SET ADDRESS OF cb2 TO PTR1(1)

MOVE cb2(97:8) TO PROGRAM-NAME2

MOVE cb2(48:1) TO JOB-CLASS

MOVE cb2(23:1) TO MSG-CLASS

MOVE ZERO TO FULL-WORD

MOVE cb2(57:3) TO FOUR-BYTES(2:3)

SET ADDRESS OF cb2 TO PTR4

MOVE cb2(25:20) TO PROGRAMMER-NAME

MOVE ZERO TO FULL-WORD

MOVE cb2(49:1) TO FOUR-BYTES(4:1)

MOVE cb2(50:FULL-WORD) TO ACCTL1

SET ADDRESS OF cb2 TO PTR1(53)

IF cb2(21:4) = LOW-VALUES THEN

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-NAME

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-NAME

PROGRAM-NAME2

JOB-NUMBER

JOB-CLASS

MSG-CLASS

DISPLAY QUOTE PROGRAMMER-NAME QUOTE

QUOTE ACCT1 QUOTE

QUOTE BATCH-OR-CICS QUOTE

USER-ID

GROUP-NAME

QUOTE USER-NAME QUOTE

GOBACK.
Output

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