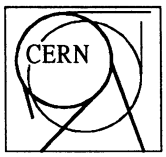
ALEPH 89-14
COMPRO 89-1
J.N. Albert
1.2.1989



ORGANISATION EUROPÉENNE POUR LA RECHERCHE NUCLÉAIRE
EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
Laboratoire Européen pour la Physique des Particules
European Laboratory for Particle Physics

Problem with the debugger under VMS 5

Jean-Noël Albert / ALEPH Online

15 December 1988

Situation

The DALI program is a software package for physics analysis written by Hans Drevermann for the ALEPH OFFLINE group. It is intended to run on VAX Station.

This program is compiled and linked with the debug options. When a user has a problem, he can stop this program using ^Y and load the debugger using the command *$ DEBUG*.

This method runs fine using VMS 4.7.

Problem from VMS 5

The first trouble arose after the upgrade to VMS 5.0, when the operation described above crashed the station. On another station, the debugger returns the error message:

%DEBUG-E-NOFREE, no free storage available

and stops. No debugger command can be sent after this message.

The first diagnostics made by the user, using repeatedly ^T, showed the problem arose when the process had used up 30000 pages of virtual memory.

Explanation

The user page fault quota has become too small with VMS 5. However, this quota is right with VMS 4.7. Increasing drastically the process quota and the system page file size enables loading of the program, but takes a very long time. Using simultaneously another process to spy the first one with *SHOW PROCESS/CONTINUE*, on a station running VMS 4.7, and on stations running VMS 5.0-1 and VMS 5.0-2, we find the following values:

VMS	Local Disk	Memory	Free Pages	Process Size	Load. Time	Node
4.7	Yes	6 Mb.	6700	16700	2 mn	AL0W18
5.0	Yes	8 Mb.	6000	34000	> 5 mn	AL0W17
5.0	No	14 Mb.	17000	34000	> 10 mn	AL0W19
5.0	No	6 Mb.	4300	34000	> 20 mn	AL0W15
5.0	Yes	5 Mb.	2500	34000	> 25 mn	AL0W13
5.0	No	5 Mb.	1700	34000	> 30 mn	AL1W02

Remark

- The station AL1W02 is a VAX Station II/GPX, diskless, running in a LAVC VMS 5. The boot node is a micro-VAX II.
- The station AL0W13 is a VAX Station 2000, with a local disk (RD 53) for the paging, running in a Mixed Cluster VMS 5. The boot node is a VAX 8700.
- The station AL0W15 is a VAX Station 2000, diskless, running in a Mixed Cluster VMS 5. The boot node is a VAX 8700.
- The station AL0W17 is a VAX Station 3200, with a local disk (RD54) for the paging, running in a Mixed Cluster VMS 5. The boot node is a VAX 8700.
- The station AL0W18 is a VAX Station 2000, running Stand Alone, with a disk (RD 54) and 6 Mb of memory.
- The station AL0W19 is a VAX Station 2000, diskless, running in a Mixed Cluster VMS 5. The boot node is a VAX 8700.

Our solution

We have found a temporary solution. We use the *VMS 4.7 DEBUGGER*.

To use the VMS 4.7 DEBUGGER, we copied the file DEBUGGER from a station running VMS 4.7 and moved it to a local directory. The standard name for the DEBUGGER is SYS$LIBARY:DEBUG.EXE. Its new name is ALEPH$SYSTEM:DEBUG_V4.EXE. So, the users can select the debugger VMS 5.0 or 4.7 defining the the logical name *LIB$DEBUG* to point to one file or the other. By default, VMS uses the VMS 5 DEBUGGER.

Example 1: Using the old DEBUGGER

```
$ DEFINE   LIB$DEBUG   ALEPH$SYSTEM:DEBUG_V4.EXE
$ RUN DA3
^Y
$ DEBUG

        VAX DEBUG Version 4.7-1

Language is FORTRAN, module to DALI1$MAIN

DBG>
```

Using this solution, we retrieve the performances of the VMS 4.7 DEBUGGER.

VMS	DEBUG	Local Disk	Memory	Free Pages	Process Size	Load. Time	Node
5.0	5.0	No	6 Mb.	4300	34000	> 20 mn	AL0W15
5.0	4.7	No	6 Mb.	4300	16869	2 mn 30	AL0W15
5.0	4.7	Yes	6 Mb.	4000	16869	2 mn 30	AL0W13
5.0	4.7	No	5 Mb.	1700	16869	2 mn 30	AL1W02

Without the debug mode, the program has approximately the same virtual size, in VMS 4.7 and VMS 5.0: less than 9076 pages.

Conclusion

The new DEBUGGER has more possibilities than the previous one. But in many case, these possibilities are not really useful.

Maybe you want to say that people have to select routines for DEBUGGER. Users should then reduce the size of the data used by the DEBUGGER. However, this is not at all convenient from the user's point of view since he can not know a priori where a problem may arise.

Using the VMS 4.7 DEBUGGER is a *temporary* solution.