

[OpenVMS] How to Troubleshoot a Process in MUTEX
Last Technical Review: 19-MAY-2000

Any party granted access to the following copyrighted information (protected under Federal Copyright Laws), pursuant to a duly executed Compaq Service Agreement may, under the terms of such agreement copy all or selected portions of this information for internal use and distribution only. No other copying or distribution for any other purpose is authorized.

Copyright (c) 1997, 2000. Compaq Computer Corporation. All rights reserved.

PRODUCT: Compaq OpenVMS Alpha, All Versions
Compaq OpenVMS VAX, All Versions

COMPONENT: Scheduler

SOURCE: Compaq Computer Corporation

OVERVIEW:

This is a general troubleshooting article for processes hung in the MUTEX wait state. See the RELATED ARTICLE section for specific troubleshooting steps on more unique issues relating to the MUTEX wait state.

QUESTION:

The DCL command SHOW SYSTEM shows one or more processes hung in the MUTEX wait state. How do you determine what the processes are waiting for, and why they are waiting?

```
$ SHOW SYSTEM
```

```
VAX/VMS V6.1 on node COORS 10-AUG-1994....
  Pid   Process Name   State  Pri
20E00401 SWAPPER          HIB    16
20E03402 wahkaw::Write  LEF    5
20E02C03 DECW$TE_2C03      LEF    6
20E00C05 SOFTBALL_ MANIAC LEF    7
20E00406 CONFIGURE     HIB   10
20E07008 J_HASSENPFIEFF    LEF    5
20E00E46 Marty          LEF    4
20E08647 BOO_BOO      MUTEX  4 <--- Process hung in MUTEX
20E00E48 Harv          LEF   16
20E04E4A Dave          LEF    4
```

ANSWER:

The operating system uses MUTEXes (Mutual Exclusion Semaphores) as a synchronization technique for shared data structures that do not require the process to be operating at elevated IPL (Interrupt Priority Level).

2. Read in the system definitions for SDA so that any MUTEX address can be interpreted.

For OpenVMS Alpha

```
SDA> READ SYS$LOADABLE_IMAGES:SYSDEF
```

For OpenVMS VAX

```
SDA> READ SYS$SYSTEM:SYSDEF
```

3. View the process on the system, noting those processes in the MUTEX state.

Note: The following is a shortcut for large systems with many processes:

```
SDA> SET OUT SUM.OUT
SDA> SHOW SUMMARY
SDA> SET OUT TT
SDA> SPAWN
$ SEARCH SUM.OUT MUTEX
```

```
SDA> SHOW SUMMARY          Current process summary
Extended Indx Process name  Username   State     Pri
-- PID --  ---
20E00401 0001 SWAPPER          SYSTEM    HIB       16
20E03402 0002 Write_Crmp      KING      LEF       5
20E02C03 0003 DECW$TE_2C03     SYSTEM    LEF       6
20E00C05 0005 BASEBALL      ROCKIE    LEF       7
20E00406 0006 CONFIGURE     SYSTEM    HIB      10
20E07008 0008 HASSENDODOO   BAYWTCH   LEF       5
20E00E46 0246 Marty           MARTY     LEF       4
20E08647 0247 BOO_BOO       HUNTER    MUTEX    4
20E00E48 0248 Harv            HOGGIE    LEF      16
20E04E4A 024A Dave            STUCKIE   LEF       4
```

4. View the process hung in MUTEX.

a. OpenVMS Alpha example

Process index: 004D Name: CHISHOLM Extended PID: 0000064D

```
-----
Process status:      02040001 RES,PHDRES,INTER
  status2:          00000001 QUANTUM_RESCHED

PCB address         80DD1440      JIB address         80E45940
PHD address         81A10000      Swapfile disk address 00000000
KTB vector address  80DD172C      HWPCB address       81A10080
Callback vector address 00000000 Termination mailbox   0000
Master internal PID 000C004D      Subprocess count     0
Creator extended PID 00000000      Creator internal PID 00000000
Previous CPU Id     00000000      Current CPU Id       00000000
Previous ASNSEQ     000000000000001C Previous ASN          0000000000000034
Initial process priority 4      Delete pending count 0
# open files allowed left 98      Direct I/O count/limit 150/150
UIC                 [00001,000004] Buffered I/O count/limit 150/150
Abs time of last event 47852C97    BUFIO byte count/limit 99552/99552
ASTs remaining      247           # of threads         1
Swapped copy of LEFC0 00000000      Timer entries allowed left 20
Swapped copy of LEFC1 00000000      Active page table count 0
Global cluster 2 pointer 00000000    Process WS page count 208
Global cluster 3 pointer 00000000    Global WS page count 48
Current capabilities: System: 0000000C QUORUM,RUN
  User: 00000000
Permanent capabilities: System: 0000000C QUORUM,RUN
  User: 00000000
Current affinities: 00000000
Permanent affinities: 00000000
Thread status:      02040001
  status2:          00000001

KTB address         80DD1440      HWPCB address       81A10080
PKTA address        7FFEFF98      Callback vector address 00000000
Internal PID        000C004D      Callback error       00000000
Extended PID        0000064D      Current CPU id       00000000
State               CUR           Flags                00000000
Base priority       4             Current priority     6
Waiting EF cluster  1             Event flag wait mask 80C064C8-->|
CPU since last quantum 0001      Mutex count         0 |
ASTs active         NONE          |
                                                    |
                                                    v
```

```
SDA> EV 80C064C8 <-----
Hex = FFFFFFFF.80C064C8      Decimal = -2134874936      LNM$AQ_MUTEX
```

```
SDA> EXAMINE FFFFFFFF.80C064C8
LNM$AQ_MUTEX: 00000000.00000001 "....."
                ^           ^
                owncnt bit 0, "write" flag
```

(This is the same quadword Mutex block when it's not owned)

```
LNM$AQ_MUTEX: FFFFFFFF.00000000 "....."
```

b. OpenVMS VAX example

```
SDA> SHOW PROCESS/INDEX=247
Process index: 0247   Name: BOO_BOO   Extended PID: 20E08647
-----
Status : 02040001 res,phdres,inter
Status2: 00000001 quantum_resched
PCB address      840BE140   JIB address      83D58DC0
PHD address      9CD08E00   Swapfile disk address 00000000
Master internal PID 00210247   Subprocess count      0
Internal PID     00210247   Creator internal PID 00000000
Extended PID     20E08647   Creator extended PID 00000000
State           MUTEX     Termination mailbox   0000
Current priority      7     AST's enabled         KESU
Base priority        4     AST's active          NONE
UIC                 [00022,000050] AST's remaining       197
Mutex count         0     Buff I/O cnt/limt    100/100
Waiting EF cluster   1     Direct I/O cnt/limt  100/100
Starting wait time   1B001B1B  BIO byte cnt/limt    65344/65344
Event flag wait mask 80004360   # open files allowed left 99
```

```
      |
      |
      +-----+
5. Translate the Event flag wait mask: |
```

```
SDA> EXAMINE 80004360 <-----+
LNM$AL_Mutex: 00010000
      ^
      bit 16, "write" flag
```

The process is waiting on the "Shared Logical Names Data Structure" MUTEX, LNM\$AL_Mutex, (see the list at the end of this article for other data structures protected by a MUTEX). The MUTEX has a single owner, i.e; Owner Count=0, who has write access to the structure, i.e; bit 16=1.

NOTE:

If the "Event flag wait mask" for the process is the same as the "JIB address", see another database article titled:

[OpenVMS] Discussion Of Unusual MUTEX Wait State

6. To see approximately how many seconds the process has been in the wait state, issue the following SDA command. The value you see may not be 100% correct due to other areas of the operating system that affect PCB\$L_WAITTIME.

```
SDA> EVAL (@EXE$GL_ABSTIM_TICS-@(PCB+PCB$L_WAITTIME))/64
```

Determining why the process is blocked from gaining access to the mutex requires that you determine which process owns the mutex. Determining the owner is difficult because there is no owner field defining this information.

When a process gains access to a mutex, its priority is raised to 16 to decrease the amount of time it has the resource. The "Mutex count" field for the process will also be incremented. Use the SDA command "SHOW SUMMARY" to determine which processes are at priority 16; those processes are possibly blocking access to the mutex (ignore the SWAPPER process, which is always at priority 16).

Isolate this list further by using the "Show Process" command, in SDA, for those suspected processes and checking to see if their "Mutex count" field is non-zero.

EXAMPLE #2:

Examining the Suspect Process

(For this example we'll use the displays from the first three commands in the previous example.)

Notice in Step 3 of EXAMPLE #1 that process "Harv" has a priority of 16 (the SWAPPER process is ignored as its priority is always 16).

1. Look at process Harv in detail, check the "Mutex count":

```
SDA> SHOW PROCESS/INDEX=248
```

```
Process index: 0248   Name: Harv   Extended PID: 20E00E48  
-----
```

```
Status : 02040001 res,phdres,inter  
Status2: 00040001 quantum_resched  
PCB address      840D2E00   JIB address      83ECC880  
PHD address      A7F54600   Swapfile disk address 00000000  
Master internal PID 00030248   Subprocess count      0  
Internal PID     00030248   Creator internal PID 00000000  
Extended PID     20E00E48   Creator extended PID 00000000  
State           LEF      Termination mailbox  0000  
Current priority      16   AST's enabled        KESU  
Base priority        4   AST's active         NONE  
UIC                  [00060,000044]  AST's remaining      197  
Mutex count          1   Buff I/O cnt/limt   99/100  
Waiting EF cluster   0   Direct I/O cnt/limt 100/100  
Starting wait time   1B001B1B   BIO byte cnt/limt   65088/65344  
Event flag wait mask DFFFFFFF    # open files allowed left 99
```

The non-zero "Mutex count" indicates that Harv owns a MUTEX, so this process is the most likely suspect to be blocking the process hung in MUTEX from gaining access to the data structure.

From this point you need to determine why this process is not releasing the mutex. However, this determination is not the scope of this article. The process is probably hung. To continue troubleshooting this problem see another article titled:

[OpenVMS] How To Troubleshoot a Hung Process

EXAMPLE #3:

Investigating Multiple MUTEX Processes

Typically, when a mutex problem occurs, it will affect more than a single process. There may also be more than 1 mutex that processes are waiting on. A single process may be blocking those processes hung in the MUTEX state, ie; "Mutex count" field greater than 1, or multiple processes may be hung and own a single mutex.

If multiple processes are hung in MUTEX and/or multiple processes have a priority 16 or higher, use SDA to produce a text file that can be searched, as opposed to using single SDA commands for each process.

Use the following commands in SDA to produce a text file for your search:

```
SDA> SET OUTPUT <filename>
SDA> SHOW SUMMARY
SDA> SHOW PROCESS ALL
SDA> SET OUTPUT TT:
```

You may now search the text file for the "Event flag wait mask" of all processes hung in MUTEX, and/or for those processes with a non-zero "Mutex count" field.

List of Data Structures Protected by Mutexes:

For both OpenVMS VAX and Alpha - longword

```
-----
```

SYMBOL	MUTEX TYPE
EXE\$GL_CEBMTX	Common Event Block List
EXE\$GL_PGDYNMTX	Paged Dynamic Memory
EXE\$GL_GSDMTX	Global Section Descriptor List
UCB\$L_LP_Mutex	Line Printer Control Block
ORB\$L_ACL_Mutex	Object Rights Block Access Control List
CHANGE_MODE_Mutex	System Service Database
TFF\$L_VEC_Mutex	Terminal Fallback Database
CIA\$GL_Mutex	System Intruder List

```
-----
```

For OpenVMS VAX -longword

```
-----
```

SYMBOL	MUTEX TYPE
LNMSAL_Mutex	Shared Logical Name Data Structures
IOC\$GL_Mutex	IO Database
EXE\$GL_SHMGSMTX	Shared Memory Global Section Descriptor
EXE\$GL_SHMMBMTX	Shared Memory Mailbox Descriptor
EXE\$GL_BASIMGMTX	Loadable Executive Image Data Structures

```
-----
```

For OpenVMS Alpha - quadword

```
-----
```

SYMBOL	MUTEX TYPE
LNMSAQ_Mutex	Shared logical name data structures
IOC\$GQ_Mutex	I/O Database
UCB\$L_SO_Mutex	Audio Device Unit Control Block
EXE\$GQ_BASIMGMTX	Loadable Executive Image Data Structures

```
-----
```