

MSM

COLLABORATORS

	<i>TITLE :</i> MSM		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 6, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MSM	1
1.1	Martins System Monitor	1
1.2	COPYRIGHT	1
1.3	REQUIREMENTS	2
1.4	INSTALLATION	2
1.5	CONTENTS	2
1.6	INTRODUCTION	2
1.7	VIRUS CHECKERS	4
1.8	STARTING MSM	4
1.9	LEAVING MSM	5
1.10	CHANGES	5
1.11	DISPLAY	7
1.12	STATUS	8
1.13	GADGETS	9
1.14	HEADER	9
1.15	MAIN DISPLAY	9
1.16	QUIT	10
1.17	PRINT	10
1.18	SAVE	10
1.19	PARAMS	10
1.20	TASK PARAM	10
1.21	USAGE	11
1.22	EXECBASE	12
1.23	GFXBASE	12
1.24	LIBRARIES - DEVICES - RESOURCES	12
1.25	PORTS	13
1.26	MEMORY	13
1.27	RESIDENT MODULES	13
1.28	EXEC FUNCTIONS	13
1.29	FONTS	14

1.30	SCREENS	14
1.31	FLUSH	14
1.32	TICK	14
1.33	FUTURE THINGS	15
1.34	BUGS	15
1.35	EXEC	15

Chapter 1

MSM

1.1 Martins System Monitor

Table of Contents

COPYRIGHT

INTRODUCTION

CONTENTS

MSM and VIRUS CHECKERS

INSTALLATION

REQUIREMENTS

STARTING MSM

LEAVING MSM

DISPLAYS

CHANGES

FUTURE ADDITIONS FOR MSM

BUGS ETC

1.2 COPYRIGHT

```

*****
*
*           M      M      SSSSS      M      M
*          MM     MM     S           MM     MM
*          M M M M     S           M M M M
*          M  M  M     SSSS      M  M  M

```

```

*           M     M           S     M     M           *
*           M     M           S     M     M           *
*           M     M     SSSSS     M     M           *
*
*           Martins System Monitor V1.9
*
*           Written by
*           Martin Ozolins
*           Copyright © 1993
*
*****

```

This program and all included files are
Copyright © 1993 by Martin Ozolins

1.3 REQUIREMENTS

MSM requires the following to operate

AmigaDOS Release 2.04 or later (V37)

An Amiga with at least a 68020 CPU (An FPU is optional)

1.4 INSTALLATION

Copy MSM and MSM.info to any directory that you like.

1.5 CONTENTS

Contents of this distribution are as follows

```

MSM           - binary executable
MSM.info      - info file for above executable
MSM.guide     - the document that you are now reading
MSM.guide.info - info file for this document

```

1.6 INTRODUCTION

What is MSM ?

MSM is a Realtime System Monitor.

That is, it constantly monitors various System Lists to
display
information on how your System is performing.

MSM is written in both Assembler and C.
The Compilers used are Devpac 3.02 and SAS/C V6.3

Although this program has been tested and so far found to be BUG Free,
Enforcer hit Free, Mungwall problem Free and IO_torture Free.
I make no gaurantee as to the fitness for use of this software.

This program is USE-AT-OWN-RISK.

If you do find any problems or bugs or just want to ask for a new feature.
I can be contacted at any of the following addresses

FIDONET 3:633/359.6
AMIGANET 41:300/359.6
TRINET 42:8699/8.6
InterNet monster@mbear.apana.org.au

This program is FREeware and may not be distributed where a cost to the
user is involved unless written permission is obtained from the author.

MSM uses an 8x8 font as opposed to a larger or smaller font for ONE reason

PERFORMANCE

Using ANY other font almost doubles the time it takes for the OS to
write the Data to the Screen.
Figures I gathered were

Approx 0.25 seconds for the 8 point Font
Approx 0.47 seconds for any other size Font

So if you would like to see MSM with a larger font, I am sorry unless there
is a large number of requests for it.

BUT, IT MAY CHANGE IN A FUTURE REVISION

Using a larger Font 9x10 also requires that the screen width be increased
by approx 30% (to fit the Data in the ListView) thus using up even more
of the precious Chip RAM.

MSM was written as I like to play around with Operating Systems.

You will notice a similarity to Xoper and MSIP, this is due to the following

MSIP is a modified version of Xoper V2.2 that I did as I was unimpressed
with the CPU figures that Xoper gave. MSIP gave CPU figures based on actual
CPU time used by a Task/Process.

MSM is a total rewrite of MSIP that uses the new gadtools.library to obtain
the now familiar 3D look.
MSM uses ListViews to display its information.

More displays and options will be added as time permits.

1.7 VIRUS CHECKERS

```
*****
*
* Warning if you are running a Virus checker program
*
* MSM SetFunctions ALL of the Exec functions ( unless you disable
* them with the ECALLS flag )
* Some Virus checking programs monitor some of these functions
* eg VirusChecker monitors the DoIO vector
*
* If you start MSM and the Virus Checker program puts up a requestor
* to inform you that a function has been changed. Click on the button
* that will NOT reset the vector as it is quite OK for this to happen.
*
* DO NOT CLICK ON THE RESTORE VECTOR BUTTON
*
* The above can happen when MSM is started AND when MSM is leaving
*
*****
```

1.8 STARTING MSM

If you use powerpacker.library I would suggest that you use loadlib to load the library prior to running MSM. The best place for this would be in your user-startup file.

The reason for this is that powerpacker.library patches FreeMem (and may also patch other EXEC routines).

This patch seems to change the way memory is freed.

As this function is patched, MSM will not be able to quit as I check to see if someone has SetFunctioned any function that I have changed.

MSM can be started from the CLI or from the WorkBench.

The Workbench accepts the following Tooltypes (Must all be in UPPERCASE)

```
STARTPRI=xxx    Used by Workbench if MSM is placed in the WBStartup drawer
TOOLPRI=xxx     Used by Workbench to set the programs priority
TIME=xxxx      Display Update Time
DISPLAY=ssssss Which display to initially show
FONT=font.name User supplied font to use other than DarkParticle
ECALLS=FALSE   Disable the SetFunctions to Exec routines
                Except the ones needed to Time Tasks
SCREEN=type     Where type equal one of the following
                LACE = An Interlace screen with the Workbench veivable height
                HIRES = Non Interlace screen
```

The CLI accepts the following Command Line Arguments using the Format of the ReadArgs from DOS V37

```
H=HELP/S        Shows a Description of Command Line options
T=TIME/K/N xxxx Display Update Time
P=PRIORITY/K/N xxx Priority at which MSM will run
D=DISPLAY/K ssssss Which display to initially show
```


- Fixed - _Backstdout was sometimes closed twice result - System Lockup
- Fixed - If we had 125 Tasks, we sometimes had a problem with cleaning up the Stack
- Fixed - If there was an Empty DIR or Filename in the Requestor. MSM would sometimes hang
- Changed - Removed animated pointers as it was found that they can cause lockups especially if Enforcer is running or under WB3
- Changed - Param windows can now be dragged around the screen
- Changed - Update time can now be as short as 1 second
- Added - Can now use the SPACEBAR to scroll down through a long display. The Keypad PAGE DN (3), PAGE UP(9), LINE DN(2) and LINE UP(8) can also be used to scroll through long displays. If you use this method to scroll, then do NOT use the mouse to scroll via the ListView scrollbar.
- Added - More Memory checking for Invalid address
Some programs DO NOT DELETE MsgPorts when they leave !!!!!.
Thus, when we try to find out the Task to Signal in the Ports display we can end up with either an INVALID RAM address or if we are lucky a VALID address with garbage in it. MSM now checks for INVALID RAM addresses and tells you if we find one.
This caused one of two problems
If you were running Enforcer - you got a hit and MSM continued on
If you were not running Enforcer - you crashed
If this happens, it would be best to re-boot your system as the Memory list may be corrupted. Better to be safe than sorry.
Programs that I know of that do this are SAS/C CPR V6.0 and 6.1 and 6.2
This has been reported to SAS/C and should be fixed in 6.3
- Added - Param windows now open beneath the Pointer
- Added - Detailed information about a Task/Process obtained from the TCB
- Added - DataSize and Address of Init Routine to Resident display
- Added - Ports Priority in Ports display

1.9

- Fixed - MSM sometimes showed over 100% CPU use upon startup
 - Fixed - MSM sometimes showed over 100% CPU if Usage button was pressed multiple times very quickly
 - Fixed - Task running time (Task Params Window) was sometimes incorrect
 - Fixed - WB3.0 - MSM crashed when showing the ExecBase
 - Fixed - Some information in the ExecBase display was incorrect
 - Fixed - Programs that used the GetCC Exec Call would crash due to how I changed this routine, this has now been fixed
AudioMaster IV was one of these programs
 - Changed - Error messages now use EasyRequestors instead of the CLI except if we cannot find intuition.library
 - Changed - Screen is not as wide - 640 pixels for now
Later versions will hopefully have a user selectable size
 - Changed - Workbench screen Height now used instead of set size
 - Changed - OSCAN_TEXT instead of OSCAN_MAX now used as some peoples system are not set up properly and thus OSCAN_MAX does not work properly
Side effect of this is that MSM uses over 30% less CPU time in drawing the display
 - Changed - If the font we get is not 8 pixels wide, we use topaz
 - Changed - Docs converted to AmigaGuide format
 - Changed - Default font is now topaz
 - Changed - Button Gadget layout with more descriptive text
-

- Changed - Task Params priority now uses a SLIDER gadget
- Changed - ExecBase and GfxBase display now show Workbench 3 info if you are using V39 or later.
- Changed - All Gadgets that can be selected via the Keyboard are now case insensitive. ie r = R (Except for the QUIT Gadget)
- Added - Screen Display now shows the Screens Depth
- Added - New Gadget - Task Params - only Enabled when we are showing the Task Control Block display
- Added - Use of the following KeyPad keys to switch between the THREE Usage displays. Update Usage, Total Usage and Tick Displays.
KeyPad 6 (arrow) and KeyPad 4 (arrow) keys to toggle
You will see either a U (for Update) or a T (for Total time)
in between the PRI and DISPAT so that you know which display is currently showing.
- Added - TICK Usage display. This shows how many times a Task has used the CPU broken into
QUANTUM
time units. Each Task is
allowed up to 4 TICKS before the Exec will Interrupt it and
schedule another Task.
- Added - ExecBase display now shows 68040 CopyBack
- Added - FLUSH command that frees Memory of un-needed libraries/fonts etc
- Added - Lots more ERROR messages
- Added - Error messages if MSM was started from the WB
- Added - FONT option to use user specified 8x8 font - default = topaz
- Added - ECalls can be disabled upon startup - new option
- Added - Uses ReadArgs for CLI options
- Added - Code to stop Q'd Msgs sometimes being missed and thus MSM was sometimes hanging
- Added - SCREEN option - can now open in Interlace or HiRes noninterlace
- Added - Param windows now open as the Active window
- Added - Task Freeze/UnFreeze
- Added - If you are using Workbench 3 or later, you can now use the mouse to scroll through the Listviews without the display jumping back to the top on each screen refresh.
- Added - Now that we use a 640 pixel wide screen, MSM should open a centered screen when your Workbench screen is larger and using OSCAN.
- Added - WB3 - uses the system standard Busy pointer

1.11 DISPLAY

MSM has 4 different display areas

STATUS

GADGETS

HEADER

MAIN

1.12 STATUS

The top ListView consists of 5 lines and is known as the Status Display ←

This Display shows the following information

```

CPU:      CPU/FFP          CPU Act:  000.000% 000.000% Idle Count:      xxxx
          Dispatches
          xxxx      Dispat/Sec: xx.x xx.x      Interrupts:      xxxx
I/O Interrupts: xxxx      I/O Ints/Sec:      xxxx.x Interrupts/Sec: xxxx.x
Update:  HH:MM:SS.micros Tasks:      HH:MM:SS.micros Idle:  HH:MM:SS.micros
Uptime:  HH:MM:SS.micros Date:      Sun 22-Nov-92 Time:      17:20:50

```

CPU: Shows the type of Processor and FFP installed

CPU Activity: Shows as a % the TOTAL amount of CPU time used
by Tasks and Processes during the Status Display
refresh time

If you have an FPU or 68040, you will notice an extra
000.000% in this area. The extra figure (the 1st one)
is the average CPU load since MSM was started

Idle Count: The number of times that the CPU had nothing to do and
thus places itself into a WAIT state to be awakened by
an interrupt. This count is only valid for each Status
Display Refresh timeframe

Dispatches: The number of times that Tasks/Processes were Dispatched
by the Exec during the Status Display Refresh timeframe.
The Total number of Dispatches that have occurred since
your system was last booted can be found in the ExecBase
display

I/O Interrupts: Number of Port interrupts during the Status Display
Refresh timeframe.

Interrupts: Number of interrupts that were processed during the
Status Display Refresh timeframe.

Dispat/Sec: The above Dispatches on a per second basis.
If you have an FPU or 68040, you will notice an extra
xxxx.x in this area. The extra figure (1st one) is the
Dispatches per second since your Amiga was last booted

I/O Ints/Sec: The above I/O Interrupts on a per second basis.

Interrupts/Sec: The above Interrupts on a per second basis.

Update: Status Display Refresh time in HH:MM:SS.micros
Can be changed by the PARAMS Gadget.

Tasks: CPU time used by Tasks/Processes during the
Status Display Refresh timeframe given in HH:MM:SS.micros.

Idle: Update time - Task time = Idle time.
This is the amount of time in HH:MM:SS.micros that the
CPU was Idle. This time includes any Exec overhead used
in Task switching etc

Just as an example

With approx 1050 Task switches per second on my
22Mhz 68030, Exec uses approx 30% of the CPU time.

Whereas with 50 - 100 Task switches per second
Exec uses approx 2 - 5 % of the CPU time.

Uptime: How long it has been since the Amiga was last booted,
shown as HHHH:MM:SS.micros

Current Date: The current date
Time: The current system time

1.13 GADGETS

These Gadgets are for the different displays that are available. Most gadgets can be accessed from the Keyboard by pressing the Key that has an Underscore below it in the Gadget

Current choices are

Quit

Print

Save

MSM Param

Task Param

Tasks

ExecBase

GfxBase

LibDevRes

Ports

Memory

Resident

ExecCall

Fonts

Screens

flush

1.14 HEADER

This is a ONE line display used as a Header line for each of the above choices.

1.15 MAIN DISPLAY

The last area (ListView) is the Main Display Area.
This is where the information is displayed.

1.16 QUIT

Quit
Leave
MSM

1.17 PRINT

Print Sends a copy of the currently displayed screen to the printer
It has been noted that it takes 2 clicks on the CANCEL requestor if
there is a printer problem. This is due to receiving the 1st CANCEL from
the actual printing and the 2nd CANCEL when we try to Close the device.

1.18 SAVE

Save Saves a copy of the currently displayed screen to a user specified file

1.19 PARAMS

Params

The MSM Param can be accessed a any time

It allows you to dynamically change the Update time in the range
of 1 to 3600 seconds

Allows you to change MSMS priority

Also shows how long MSM has been running

1.20 TASK PARAM

The Task Param can only be used if you are looking at the
Task Control Block of a Task

With this window, you can change a Tasks Priority, see how long
it has been since the Task started and you can also Freeze/UnFreeze
the Task

1.21 USAGE

Usage Information about Tasks/Processes that are loaded
This Display has 3 different sub displays accessed via the Left and Right
arrow keys on the numeric KeyPad

Usage and Total show the following information

```

ADDRESS          Address of Task Control Block
STATE            The State of the Task/Process - Active Ready Waiting etc
PRI              Priority of Task/Process
U or T          Stands for USAGE display and TOTAL time used display
DISPAT          If U is displayed
Number of times that this Task has been Dispatched by
Exec during the Main Display Refresh timeframe.
If T is displayed
TOTAL NBR of Dispatches of the Task
  HH:MM:SS.micros If U is displayed
CPU time used by the Task during the Main Display
Refresh time.
If T is displayed
Total CPU time used by the Task
  %CPU           If U is displayed
The above expressed as a % based on the Main Display
Refresh time.
If T is displayed
CPU% that the Task has used since recording was started
or since the Task was started
NAME            The name of the Task/Process

```

The Header line also shows how many tasks/processes are loaded along
with the Main Display Refresh time (Not shown in TOTAL display).

Tick display shows the following

```

ADDRESS          Address of Task Control Block
STATE            The State of the Task/Process - Active Ready Waiting etc
PRI              Priority of Task/Process
TICK0           Number of times that the Task/Process used less than
ONE
                Tick
                TICK1   Number of times that the Task/Process used between
ONE and TWO Ticks
TICK2           Number of times that the Task/Process used between
TWO and THREE Ticks
TICK3           Number of times that the Task/Process used between
THREE and is FULL
                Quantum
                Time Slice
NAME            The name of the Task/Process

```

All THREE Displays can do the following

If you click on a Task/Process you will see information about the Task.
Stack size and Stack used
The whole Task structure is displayed

If it is a Process, the Process structure is also shown.
 This includes SegLists and Total SegList size
 If you now bring up the Params gadget you can change
 the Tasks Priority.
 Task running time is also displayed.

1.22 EXECBASE

EBase A complete and commented description of the current state of
 the ExecBase along with which flags are set in the MMU if fitted.
 This display Auto adjusts for WB2 and WB3

1.23 GFXBASE

Gfx A detailed and commented listing of GfxBase
 This display Auto adjusts for WB2 and WB3

1.24 LIBRARIES - DEVICES - RESOURCES

LDR The Libraries, Devices and Resources that are currently loaded
 into memory.

ADDRESS	Base Address
Checksum	The system calculated checksum
NegS	Number of bytes before the LIBRARY.
PosS	Number of bytes after LIBRARY.
Flag	0 = system is currently checksumming 1 = library changed since last checksum 2 = library allows checksumming 3 = delayed expunge flag 4 = special system expunge flag
Ver	Version (UWORD)
Rev	Revision number (UWORD) You may see a very high Ver or Rev number on some Libraries, Devices or Resource, this would be due to the fact that I check the WHOLE WORD and not just the BYTE that holds this information. If I checked just the byte, the number would seem to be correct, but why should I cover up the mistake, the field is a WORD so that is what I check.
Open	Number of current opens (UWORD)
Name of Library Device Resource	

Also shown is the number of each type loaded

1.25 PORTS

Ports A list of Public Ports in the system

 ADDRESS Address of Port structure

 FLAGS SIGNAL ACTION SOFTINT IGNORE

 SigBit signal bit number

 Priority Priority of Port

 Msgs Number of messages the Port is waiting on

 Name of PORT

 Object to Signal

 The task/process that gets signalled
 when a message arrives

1.26 MEMORY

Mem Shows the different types of Memory configured.

 Lower Lowest address that is available for use

 Upper Highest address available for use

 First Free First Free region

 Available Number of bytes available for the type.

 Attr Characteristics of this region.

 Pri Priority of the configured memory.

 Name of the Memory Region

1.27 RESIDENT MODULES

Res Display the Resident Modules

 ADDRESS Address of the Resident Module

 Ver Version of the Module

 Type Library Resource Device Unknown

 Flags

 Pri Priority of the Module

 InitCode

 EndSkip

 DataSize

 InitRout

 Module Name of the Module

1.28 EXEC FUNCTIONS

ECall Displays ALL of the EXEC Functions (as of V37)
 and how many times they have been called.

 There are 2 figures

 TOTAL since MSM was started

 UPDATE since the last EXEC display update

 There is also a count of the Total calls to the

Exec Functions since MSM was started and a Total
calls to the Exec Functions since the last update
of the EXECC display

1.29 FONTS

Font A listing of the Fonts that are currently loaded
 into Memory

ADDRESS Address of Font
Height The Height of the Font in pixels
Width Width of the Font in pixels
Style Flags that indicate whether it is BOLD etc
BaseL The BaseLine of the Font
BoldS BoldSmear
Cnt Open count of the FOnT
Name Name of the Font

1.30 SCREENS

Scrn List of Screens and there Windows

Title of the Screen
Default Title for the Screen
The Screens default Font
LeftEdge, TopEdge, Width, Height, Screen Flags and Depth

The Windows Height, Width, TopEdge, LeftEdge and its Flags

Total number of Screens and Windows that are open

1.31 FLUSH

flush Equivelant to running "AVAIL FLUSH" 3 times
 Forces an Unload of Libraries/Fonts which are not in use

1.32 TICK

A Tick is 1/4 of a Time Slice Quantum - in previous releases (pre V36) a
Tick was 1/6 of a Time Slice Quantum)

A Quantum is equal to ONE Screen refresh

1.33 FUTURE THINGS

Things I want to add for future MSM (not in any special order)

- 1 - Seperate display to show Task/Process stacks and Register Values
- 2 - Support different Screen modes
- 3 - Open its Window on the Workbench Screen
- 4 - Add a HotKey so that it runs in the Background and will open its Screen when the Hotkey is pressed
- 5 - Display to show Interrupts
- 6 - Reset the Time and CPU counts of a Task via user request
- 7 - Close Un-used windows/Screens
- 8 - User suggestions ?
What other displays do you want ?

1.34 BUGS

Currently there is ONE known sneaky bug that only happens on some machines. It does not happen on mine and mainly affects Amigas with a 68040. Sometimes the System SOMETIMES will hang if you use the PORTS display. What is even more strange is that if you load MSM and use the ports Display and the System hangs, quite often it will not hang again. It is very strange and annoying. TWO options have been added to help me find out what is causing this. PORTDIS and PORTDIS1. If this happens to you, try using the PORTDIS option, if it still happens try the PORTDIS1 option. Either way, can you let me know what happens if you use either/both of these options ?

If you do have this happen to you, can you send me a list of what was running on your system and a description of your system containing your Hardware and Software configuration

Send ALL Bug reports to
bugs@mbear.apana.org.au

and ALL suggestions to
suggestions@mbear.apana.org.au

1.35 EXEC

The Amiga Operating System is a MultiTasking Pre-Emptive Operating System. As such, it decides which Task/Process will have control of the CPU. How does it know when to allow another Task/Process access to the CPU ?

The Exec has THREE lists which keep track of the Tasks

The FIRST list consists of ONE Entry
The Task that is currently in control of the CPU

The SECOND list is a list of Tasks that are READY to use the CPU

These could be CPU intensive Tasks
They could be Tasks that have received a Signal
(whilst in the WAIT list)
and have been now moved into the READY queue

The THIRD list is a list of Tasks that are WAITing for a Signal to wake them up
The Signal could be a Keypress, a Mouse event, a reply from a Disk I/O
or virtually anything

When a Task has used up ALL of its Quantum Time Slice or is WAITing on a Signal
The EXEC will look through its READY list to see which Task has the Highest
priority.

If it finds a Task with a priority that is higher than the priority of the Task
that is losing access to the CPU, this new Task will be given control of the CPU.
If there are NO Tasks with a higher priority, one of the following will happen.

If the Task that is losing the CPU lost control due to its Quantum Time
Slice being fully used up it will again be given control of the CPU
provided that NO other Task in the READY queue has an EQUAL priority.
If another Task in the READY queue has the same priority, then this new
Task will be given control of the CPU.

Thus, a task with a lower priority will not be given control of the CPU.
If no Task has a higher or equal priority, then a lower priority Task
will be given CPU control

If the Task lost control because it is WAITing on a Signal.
The highest priority Task on the READY queue will be given CPU control.
This Task could have a lower priority than the Task that just lost
control.

Another reason why a Task could lose control of the CPU is when a Task on the
WAIT queue receives a Signal. The Exec then goes through its Interrupt routine
and looks through the READY list. If it finds a Task with a higher priority than
the currently active Task, it will allow this new Task control of the CPU
(Pre-Emptive Task Switching)

The worst way that a Task can lose control of the CPU is to visit Mr GURU