

doc/VMM40

COLLABORATORS

	<i>TITLE :</i> doc/VMM40		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 29, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	doc/VMM40	1
1.1	VMM40/doc/VMM40.guide	1
1.2	VMM40/doc/VMM40.guide/COPYRIGHT	2
1.3	VMM40/doc/VMM40.guide/INTRODUCTION	2
1.4	VMM40/doc/VMM40.guide/INSTALLATION	2
1.5	VMM40/doc/VMM40.guide/VMM40PREFS	3
1.6	VMM40/doc/VMM40.guide/REGISTRATION	4
1.7	VMM40/doc/VMM40.guide/PROBLEMS	4
1.8	VMM40/doc/VMM40.guide/QUESTIONS	5
1.9	VMM40/doc/VMM40.guide/TECHNICAL_DES	5
1.10	VMM40/doc/VMM40.guide/KNOWN_BUGS	7
1.11	VMM40/doc/VMM40.guide/FUTURE_PLANS	7
1.12	VMM40/doc/VMM40.guide/ACKNOWLEDGMENTS	7
1.13	VMM40/doc/VMM40.guide/MISCELLANEOUS	8

Chapter 1

doc/VMM40

1.1 VMM40/doc/VMM40.guide

```
VMM40
(Virtual Memory Machine for Amigas with 68040)
User's Guide
Version 1.0
$Date: 93/12/07 19:00:06 $
written by Martin Apel
email: apel@gypsy.physik.uni-kl.de
```

CONTENTS

- 0.
- Copyright
 - 1.
- Introduction
 - 2.
- Installation
 - 3.
- VMM40Prefs
 - 4.
- Registration
 - 5.
- Problems
 - 6.
- Questions and Answers
 - 7.
- Technical description
 - 8.
- Known bugs
 - 9.
- Future plans
 - 10.
- Acknowledgments
 - 11.
- Miscellaneous

1.2 VMM40/doc/VMM40.guide/COPYRIGHT

IMPORTANT NOTICE: This program is copyrighted by Martin Apel, but can be freely distributed, provided that the following rules are respected.

- No change is made to the program nor to the accompanying documentation
- The package is always distributed in its complete form consisting of the following files: "VMM40", "StartVMM40", "VMM40Prefs", "VMM40.guide", "TestedPrograms", "RegistrationForm", "VMM40.prefs", "VMProgGuideline", "README" and "History".
- Every form of distribution is allowed and encouraged, but no fee can be charged for this program except for, possibly, the cost of magnetic media and/or disk duplication and shipping.
- Inclusion in PD software libraries such as Fish Disks is allowed, provided the fees charged for these disks are comparable with those charged by Fred Fish.
- The program cannot be distributed in any commercial product without the written consent of the author.

By copying, distributing and/or using the program you indicate your acceptance of the above rules.

1.3 VMM40/doc/VMM40.guide/INTRODUCTION

1. INTRODUCTION

Even on the A4000 equipped with 6 MB sometimes I longed for more memory or, as an alternative, for virtual memory. As the 68040 contains an MMU and I was interested in learning how it works, I decided to write a virtual memory manager for the Amiga myself. It emulates up to 32 MB in a user selectable amount of physical RAM. In this first version paging is only supported to partitions, not to files. So in order to use VMM40 you have to provide a partition of the size you need.

VMM40 installs a standard memory list in ExecBase, so virtual memory will be handled just like other memory. VM is allocated only when the MEMF_PUBLIC flag in the allocation is not set. Otherwise system data such as task control blocks and IORequests might be paged out, which would lead to failure. Code can NOT be put into virtual memory, because the DOS loader always requests memory with MEMF_PUBLIC flag set.

1.4 VMM40/doc/VMM40.guide/INSTALLATION

2. INSTALLATION

VMM40 requires at least OS2.0 to run. To install VMM40 you simply have to copy the executables into a directory in your path. All these files have to be in the same directory. The first time you use VMM40,

you have to start VMM40Prefs first to specify the paging partition, the amount of RAM dedicated to paging and so on. To start VMM40, simply call StartVMM40 without any parameters. To quit, start StartVMM40 again, or click the close gadget of the statistics window, if statistics are enabled. A sample configuration file "VMM40.prefs" is provided which you can use as a starting configuration (paging partition still has to be entered).

1.5 VMM40/doc/VMM40.guide/VMM40PREFS

3. VMM40PREFS

In order to enter all the settings that are needed for VMM40 to work, there's a nice GUI provided with the program. Do NEVER modify the configuration file generated by VMM40Prefs by hand. If this file is not what it's supposed to be, VMM40 may crash or, even worse, use the wrong partition or the like.

Here's a description of the gadgets and what they do:

Tasks / Programs list:

You can enter task or program names, which should always be run either using VM or not. They are entered with the state of the "Use VM" gadget below the list and can be modified afterwards. The name you enter can either be a task name, the name of a load file (without path), or even a normal AmigaDOS pattern to specify which entity is meant to be run with or without VM.

VM by default:

If selected, all programs not found in the task list may use VM, otherwise not.

Statistics:

If selected, a small window is opened, which displays statistics of the paging process. It contains a close gadget. If you click this AND no more VM is currently in use, VMM40 will quit. If there's any VM still in use, VMM40 will not allocate any more VM to other requesting tasks, but will quit as soon as the rest of VM is returned.

The memory allocation gadgets:

The first gadget determines, if VMM40 uses either the largest continuous chunk of memory available at startup for paging or a selectable amount, which you can select with the slider gadget just below.

Swap medium:

Until now only paging to a partition is supported. When you click on "Select" a window with all drives will show up and you can select one of them. If you select a partition for paging for the first time, VMM40 will ask you if you really want to overwrite that partition, so you need not worry about destroying the wrong partition.

Swap file size:

This gadget is only valid for paging to a file. As this is not yet

implemented, it is always disabled for now.

VM priority:

Here you can select, when VM will be allocated. Exec searches through all memory lists in a priority order with normal FAST memory usually at priority 30, CHIP at -10. So if you want to allocate VM first, you have to select a value larger than 30; if you want to use normal fast memory first, but VM before CHIP memory, you have to select a value between -10 and 30.

1.6 VMM40/doc/VMM40.guide/REGISTRATION

4. REGISTRATION

This program isn't shareware or the like, but I wish you would send me your configuration data, so I can see potential problems with certain configurations. VMM40 has been thoroughly tested on a A4000/40 with 6 MB and the standard IDE-harddisk used as a paging device. I have included a registration form in this distribution I want you to send me, if you are using VMM40 successfully or you are experiencing any problems with it.

1.7 VMM40/doc/VMM40.guide/PROBLEMS

5. PROBLEMS

Commodore defined the MEMF_PUBLIC flag for the AllocMem function once upon a time, when no-one knew what this would mean in future days. The result is that either people allocated all their memory with the MEMF_PUBLIC flag set or they ignored this at all and never set it. The first way doesn't hurt VMM40, but it makes the corresponding program not use virtual memory. The second alternative is worse. There are a lot of programs, which allocate messages or IORequests on the stack, which might produce a failure in a virtual memory environment. Such programs can be forbidden to use virtual memory using the preferences program, otherwise spurious crashes may result. If you are considering writing programs which might benefit from virtual memory, you should read the file "VMProgGuideline" to see what should be avoided in a virtual memory environment.

The other major problem is that I don't know whether VMM40 works on machines other than the A4000. I don't have access to an Amiga with a 68040 card installed, so testing is quite difficult. I hope this will be fixed in the next release.

Caching programs such as FastCache or PowerCache should be run after VMM40. They should also be disabled from using virtual memory (doesn't make sense to give virtual memory to a caching program, does it?). The same applies to all programs that patch the BeginIO vector of the paging device.

Programs which use the Access Fault Trap Vector such as Enforcer have to be run before VMM40, otherwise Enforcer will flag all page-faults

as invalid memory references.

1.8 VMM40/doc/VMM40.guide/QUESTIONS

6. QUESTIONS AND ANSWERS

Question: Program "X" doesn't use virtual memory. Why?

Answer: "X" might always allocate memory with the MEMF_PUBLIC flag set. There's currently no way to make "X" use virtual memory. If this is not the case, you may have excluded it from using virtual memory in VMM40Prefs.

Question: VMM40 crashes with my configuration. How do I find out which program behaves badly ?

Answer: Disable the gadget "VM by default" in VMM40Prefs and move the memory slider to a low position, so there are many page-faults, when VMM40 runs. Then enable virtual memory for each task in your system one after another. This way you can see, when the system crashes first. You can disable VM for the task producing the crashes then.

1.9 VMM40/doc/VMM40.guide/TECHNICAL_DES

7. TECHNICAL DESCRIPTION

VMM40 consists of three tasks, enabling the statistics window invokes a fourth. The first one is the VM_Manager, who takes care of deciding which task is allowed to use VM, initializing everything and so on. The PageHandler does the paging, when a page-fault has occurred. The Prepager causes pages to be locked in memory, when IO has to be carried out to or from VM using the paging device. Most of the effort has not been invested into getting the paging to run, but to make the system use VM correctly under all circumstances. Unfortunately Commodore hasn't invested much thought into the definition of the MEMF_PUBLIC flag, so a few system functions had to be patched. I have tried to keep VMM40 as system-friendly as possible, but I had to make a few assumptions, which are not documented. The worst one is that I had to patch the Switch function, which puts a task from running into the ready or wait state. This means that VMM40 might not run in future versions of the OS, though I don't think Commodore will be changing much in such low level code.

Commodore also didn't state anything about using non-public memory inside Forbid/Permit or Disable/Enable. Causing a page-fault inside a forbidden/disabled section is dangerous, because paging results in task switching. Allocating virtual memory inside a forbidden/disabled section is consequently disabled, the same applies to freeing memory. Memory which should be freed inside such a section is freed when this section is completed.

Currently the number of faults in process plus the number of tasks using VM for their stack must not be more than 20 at any moment.

The VM_Manager process

The VM_Manager starts up all other tasks and initializes most data. It also handles the quit request by the user. Each time AllocMem is called, VMM40 has to decide whether the requesting process is allowed to use VM. The first time a task/process allocates memory, a message is sent to the VM_Manager, which decides if VM is permitted. Subsequent requests by the same task are decided using a hash table.

The PageHandler task

All paging is handled asynchronously by the page handler. When a fault occurs, the params for such a fault are put into a so-called trap-struct by the trap-handling code and this is sent to the pagehandler. The pagehandler chooses a page to be evicted and possibly writes out a modified page out first. When this process is finished, the required page is read in. When the read has been successfully completed, the faulting task is signalled to continue its computation. During IO to the paging device other tasks are permitted to run and eventually cause other page-faults, which are handled immediately.

The Prepager task

In the process of writing VMM40 I have detected some cases, in which IO to the paging device is requested, e.g. to another partition living on the same physical device as the paging partition. As the paging device itself must never block for a page-fault, this has to be prevented. All IO to the paging device (except that from the pagehandler) is examined for usage of virtual memory. If it uses VM, the corresponding pages are locked into memory, the IO is carried out, the pages are unlocked and the process requesting IO is permitted to continue.

The statistics task

The statistics task is only created, when statistics are enabled using the VMM40Prefs program. Every second it prints a few lines about usage of virtual memory, number of page-faults occurred so far and the number of trap-structs currently in use.

Patches to system functions

The following Exec functions are patched by VMM40: Switch, AddTask, AllocMem, FreeMem. Additionally the BeginIO function of the paging device is patched. The Switch and AddTask functions had to be patched because the stack may be in virtual memory. To prevent page-faults while in supervisor mode (task-switching), the stack is replaced by a temporary stack large enough to contain all registers pushed onto the stack during a context switch. When the task is re-launched, the original stack is used again.

1.10 VMM40/doc/VMM40.guide/KNOWN_BUGS

8. KNOWN BUGS

As far as I know there are no major bugs in VMM40. However there is one minor bug which still has to be corrected. If a task is removed via RemTask from another task and it uses virtual memory for its stack, the TrapStruct is currently not released. Commodore recommends not to call RemTask on another task and it isn't done very often in real programs. One possibility is to prevent the task that allocates the stack for such programs from using virtual memory.

1.11 VMM40/doc/VMM40.guide/FUTURE_PLANS

9. FUTURE PLANS

There are several features I want to include in future versions of VMM40 such as:

- Paging to a disk file
- Dynamic allocation and freeing of memory used for paging
- Dynamic allocation of TrapStructs (see
 7. Technical description
)
- Support to use virtual memory for code segments / seglists (maybe)

I would like to include a list of working programs and of programs with problems in this documentation in the next release (based on your registration data).

1.12 VMM40/doc/VMM40.guide/ACKNOWLEDGMENTS

10. ACKNOWLEDGMENTS

I would like to thank the following people for doing the Beta testing and providing me with ideas:

Torsten Stolpmann
Juergen Zimmermann
Torsten Ebeling

1.13 VMM40/doc/VMM40.guide/MISCELLANEOUS

11. MISCELLANEOUS

I would be glad to hear from you, if VMM40 works on your machine, what programs have difficulties in running together with VMM40. Please mail the filled in registration form to my email address:
apel@gypsy.physik.uni-kl.de.
