

memory

Marius Gröger

Copyright © CopyrightÂ©1996 Marius Gröger, All Rights Reserved.

COLLABORATORS

	<i>TITLE :</i> memory		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Marius Gröger	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	memory	1
1.1	Memory-Device	1
1.2	Memory-Device - Description	2
1.3	Memory-Device - Feaures	2
1.4	Memory-Device - Requirements	2
1.5	Memory-Device - Installation	3
1.6	Memory-Device - Example for VMM usage	4
1.7	Memory-Device - MakeDisk	5
1.8	Memory-Device - Known Bugs	6
1.9	Memory-Device - History	6
1.10	Memory-Device - Copyright / Credits / Legal stuff / Contacting	7

Chapter 1

memory

1.1 Memory-Device

Memory-Device -- Exec device driver for direct memory access

(C) Copyright 1996 Marius Gröger.

WARNING: THIS SOFTWARE SHOULD ONLY BE INSTALLED AND USED BY EXPERIENCED
USERS! IT IS A LICENSE TO CRASH THE SYSTEM!

TABLE OF CONTENTS

Intro

Description
Why You need it!

Features
All in one glance

Requirements
What you need

Device Operation

Installation
What's needed to be done

Example with VMM
How to use it with VMM

MakeDisk
The RAM disk generator

Bugs
Known Bugs

Appendices

History
 What's been done already

Copyright / Author
 Legal stuff, credits, my addresses

1.2 Memory-Device - Description

Description

~~~~~

What is memory.device?

This special device simulates a trackdisk compliant block device in any region of memory. Unlike the RAM DISK (which is in fact not even a device) or the ramdrive.device, memory.device allows the exact definition of the memory to use for data storage.

The purpose memory.device has been developed for is to use it on systems equipped with different RAM chips or local memory busses. In conjunction with VMM or similar virtual memory packages, a memory region with slower RAM chips may be used as swap area. This idea originates from the z2ram device which has been written for Linux/68k/Amiga.

I tried to use the MEM-handler before writing memory.device, but it seems that VMM actually needs a real Exec device, even when told to use a plain page file. I believe nobody has ever been struck by this because probably nobody ever tried to swap to RAM:pagefile :-)

Other applications with memory.device are conceivable, ranging from whitty, over funny to perverted. Use it on your own risk.

## 1.3 Memory-Device - Feaures

### Features

~~~~~

- Very fast raw memory access
- Compatible with trackdisk's command set
- Can be used as RAM disk, see
 MakeDisk
 - ~Distributed as Freeware, complete source code is supplied

1.4 Memory-Device - Requirements

Requirements

~~~~~

- AmigaOS Release 2.04 or better
- ~VMM if you want to use memory.device for swapping

## 1.5 Memory-Device - Installation

## Installation

~~~~~

1. Copy the file memory.device to the system devices directory DEVS:
2. Copy the AmigaDOS Driver Descriptor MEMORY to DEVS:DOSDrivers/
3. This step should be performed with maximum attention. Any mistake will in 99% of all cases lead to an unpleasant system shutdown.

Edit the AmigaDOS Driver Descriptor to match your needs. It is not important how you organize the geometry of the device, only the SectorSize should not be too odd. Probably it's best to stick with a value of 512 bytes. Choose the values to match the following equations:

$$\text{LowCyl} \times \text{Surfaces} \times \text{SectorsPerTrack} \times \text{SectorSize} == \text{Starting-Address} \leftarrow$$

$$\text{SectorsPerTrack} + \text{HighCyl} \times \text{Surfaces} \times \text{SectorsPerTrack} \times \text{SectorSize} == \text{Ending-Address} \leftarrow$$

This leads to:

$$\text{LowCyl} := \text{Starting-Address} / (\text{Surfaces} \times \text{SectorsPerTrack} \times \text{SectorSize}) \leftarrow$$

$$\text{HighCyl} := (\text{Ending-Address} - \text{Starting-Address}) / (\text{Surfaces} \times \text{SectorsPerTrack} \times \text{SectorSize}) \leftarrow$$

When dealing with RAM addresses, take care of not to confuse Hexadecimal and decimal addresses.

4. Depending on your application the most tricky bit might be the task of convincing the operations system (namely exec.library or expansion.library) not to configure a certain memory region to be used. For instance, a GVP Series II RAM expansion needs to be configured by expansion.library, otherwise the RAM won't even appear on the bus.

However, once configured by expansion, any RAM will also be added to Exec's memory lists. There are several tools available on Aminet which seem to deal with RAM expansion boards.

One way might for example be to configure the RAM (e.g. by hardware jumpering) not to auto-configure at boot-time. In startup-sequence, an `addmem` command or similar could configure the RAM, and an `AllocAbs()` following right after this could eat the complete new block.

1.6 Memory-Device - Example for VMM usage

An example to use `memory.device` as VMM swap space

~~~~~

Important Note: Before you actually start toying around with VMM, you should have installed the `memory.device` and found a working method to hide some of your RAM from the system. It is most important that the region of memory you intend to use with `memory.device` is either completely unconfigured or completely allocated in an Exec-safe manner. It is impossible to describe how to archive this on your system, you simply must have the required knowledge about the Amiga's internal way of memory and expansion handling. See

Installation  
for further detail.

The idea of it

`Memory.device` has been developed to be used as swap space with VMM. This is realised in a similar way as done in Linux/68k/Amiga's `z2ram` device.

Usually any harddisk partition is driven by some `scsi.device` which provides for the low-level device functions, namely to read and write blocks of data. The idea is just to pretend the slow RAM portion being a very small harddisk, and VMM won't notice anything.

Now let's get into the details.

System examination

A typical configuration is an A2000 system with a A2630 turbo board providing 4 megs 32bit RAM, and another 4megs on a 16bit RAM expansion (such as a dedicated RAM board, or a SCSI host adapter with a RAM option).

First you should figure out the memory map to avoid sacrificing innocent memory. The memory map of the above configuration looks like this:

```

+-----+-----+---$0000 0000
|   512k Chip RAM   |
+-----+-----+---$0080 0000
| 512k Chip RAM (opt.) |
+-----+-----+---$0100 0000
|   1meg Chip RAM   |
|   (opt.)          |
+-----+-----+---$0200 0000
|                   |
|       4megs of    |
|   fast 32bit RAM  |
|                   |
+-----+-----+---$0600 0000 \

```



```

|           |           |
|   4megs of |           | Should be used
| slow 16bit RAM |         | for swapping
|           |           |
+-----+-----+---$0A00 0000 /
|           |           | | | | | | |
| IO and ROM follows |         |
|           |           |
| | | | | | | | |

```

## 2. The DOSDriver entry

The DOSDriver entry ``MEMORY'' must be modified to meet the above considerations. The following values should (at least for calculation simplicity) always chosen like this:

```

Surfaces := 1
SectorsPerTrack := 1
SectorSize := 512

```

Low- and High-Cylinder will determine the actual RAM area:

```

LowCyl := $600000 / (1 $\times$ 1 $\times$ 512) = 12288
HighCyl := ($A00000-512) / (1 $\times$ 1 $\times$ 512) = 20479

```

Note that both hexadecimal and decimal values are used.

## 3. VMM configuration

Finally start the VMM configure utility and put VMM in the mode ``Swap to Partition''. Enter ``MEMORY'' in the text-field which controls the partition to be used.

VMM should soon start swapping, since half of your RAM seems to be vanished for your system.

## 1.7 Memory-Device - MakeDisk

The ``MakeDisk'' tool

```

~~~~~

```

The MakeDisk tool serves to create a RAM disk based on the memory.device. Such a RAM disk has the following advantages compared to the standard RAM Disk (ram-handler) and RAD (ramdrive.device).

- Very few overhead
- You may specify if chip RAM should be used
- ~No need to write a Mountlist or 2.1 style DosDriver entry

The disadvantages are

- Static size, will be always 1K to less
- Static size, can't be shrunk by just deleting some file
- ~Can't be dismounted

If you want a more flexible solution, use some other RAM disk, there are a lot flowing through the Net. I definitely won't support MakeDisk very much, for me it suits fine (BTW: I keep my ENV: in this RAM disk, as my fast RAM is too precious).

The template of MakeDisk is

```
DRIVE/A,NAME/K,B=BLOCKS/K/A,C=CHIP/S
```

DRIVE -- Name of DOS device without trailing colon (:).

NAME -- Name of formatted volume without trailing colon (:).

BLOCKS -- size of drive in blocks. Each block holds 512 bytes.

CHIP -- RAM should be allocated from chip RAM.

Example:

```
AmigaShell> MakeDisk DRIVE="CHIPENV" NAME="Chip-Env" BLOCKS=300 CHIP
```

This creates a DOS drive CHIPENV: with the volume name Chip-Env. The size of the drive is about 150 KBytes, the RAM is allocated from chip RAM.

## 1.8 Memory-Device - Known Bugs

Known Bugs

~~~~~

Currently things seems to work quite well.

If you know bugs, contact  
me

.

## 1.9 Memory-Device - History

Development history of SANA2Meter

~~~~~

Release 37.4

- trackdisk emulation improved, now suitable for RAM disk usage
- MakeDisk tool for RAM disk creation

Release 37.3

- missing DOSDriver included

---

- minor changes

Release 37.2

- Faster block copy routine
- support for CMD\_UPDATE

Release 37.1

- ~initial release

Technical history of Memory-Device

For more technical details, please refer to the source code. In the header section of each source file, you find the pseudo-keyword \$HISTORY: after which follows the complete revision history of this file.

## 1.10 Memory-Device - Copyright / Credits / Legal stuff / Contacting

Copyright, Credits and Author Information

~~~~~

Credits

The technical idea behind the memory.device was inspired by the z2ram device written for Linux/68k/Amiga.

Legal stuff

memory.device

(C) Copyright 1996 Marius Gröger. All Rights Reserved.

This software is freeware. It is provided as-is and is subject to change; no warranties are made. All use is at your own risk. No liability or responsibility is assumed.

It's strictly forbidden to include this archive in any kind of software collection except Fred Fish's AmigaLibrary, Meeting Pearls CD's, Aminet, Aminet CD's and BBS fileareas.

You may make any changes to the source for your own use. If you consider them useful for everybody, please get in touch with me. This way I can try to include them in the next public release.

AmigaGuide, AmigaGuide.info, amigaguide.library, WDisplay, WDisplay.info  
(c) Copyright 1991-93 Commodore-Amiga, Inc. All Rights Reserved.  
Reproduced and distributed under license from Commodore.

Contact

If you like this software, or have any suggestion how to improve it, or just want to complain about it, feel free to email me.

## Home address:

Marius Gröger,  
Bärstadter Str. 4  
65307 Bad Schwalbach (GERMANY)

## Internet email addresses:

mag@sysgo.de  
groeger@goofy.zdv.uni-mainz.de

---