memory

Marius Gröger

COLLABORATORS

	TITLE : memory		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Marius Gröger	July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

memory

1.1 Memory-Device

Memory-Device - Exec device driver for direct memory access - Written by Marius ↩
    Gröger.

This software requires AmigaOS 2.0 (V37) or higher.

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

```
    Description          Why You need it!
    Features             All in one glance
    Requirements         What you need

    Installation         What's needed to be done
    Example with VMM     How to use it with VMM
    Bugs                 Known Bugs

    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
quipped 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
the VMM actually needs an underlying Exec device, even if using 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.

The memory.device is in fact a very small driver, and writing this
documentation was really a harder task. The good side of this is that the
device operates at a very high speed. For example, the device's read and
write functions are hardly more than a simple call to Exec's fast memory
copy routine. All other typical device functions are simply stubs to
pretend a correctly operating, writable trackdisk device.

1.3   Memory-Device - Feaures

Features

· Very fast raw memory access

· Compatible with trackdisk's command set

·~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

```
    equastions:

    LowCyl $\times$ Surfaces $\times$ SectorsPerTrack $\times$ SectorSize ==  ←
        Starting-Address

    SectorsPerTrack + HighCyl $\times$ Surfaces $\times$ SectorsPerTrack $\times$ ←
        SectorSize ==
                                                      Ending-Address

    This leads to:

    LowCyl := Starting-Address / (Surfaces $\times$ SectorsPerTrack $\times$  ←
        SectorSize)

    HighCyl := (Ending-Address-SectorsPerTrack) /
                                    (Surfaces $\times$ SectorsPerTrack $\times$  ←
                                SectorSize)

    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 methnod 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 completetly
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 sacrifycing 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 - Known Bugs

Known Bugs

Currently things seems to work quite well.

If you know bugs, contact me .

1.8   Memory-Device - History

Development history of Memory-Device

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.

1.9   Memory-Device - Copyright / Credits / Legal stuff / Contacting

Copyright and author information

Memory-Device is

(C) Copyright 1995 Marius Gröger
    All Rights Reserved.

Credits

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

I ask everyone who has suggestions to Memory-Device's implementation or
even modified versions of Memory-Device to tell me, so that I can
incorporate them into the distribution.

Legal stuff

Contact

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

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

Internet email addresses:
   mag@sysgo.de
   groeger@gundel.zdv.uni-mainz.de
   i409@informatik.fh-wiesbaden.de