MuFastZero

Thomas Richter

COLLABORATORS			
	TITLE : MuFastZero		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Thomas Richter	April 14, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

MuFastZero

1.1 MuFastZero Guide

´### ,#. ,#### ,### ####’ #### ,#### ,###’ ####’ #### _____ ,#### ,###’ || ####’ #### | ___ ___ | ___ ,#### ,###’ ---- | | | | | | |___ ####’ #### | | | | | | | ,#####. ,###’ . ||___||___||_ ___| ####’##. ,#### ,## ,#### ########’ # ,##’ ####’ `####’ `###’ ,#### ##### © 1999 THOR - Software, ####’ Thomas Richter `##’

MuFastZero Guide

Guide Version 1.01 MuFastZero Version 40.3

The Licence : Legal restrictions

MuTools : What is this all about, and what’s the MMU library?

What is it : Overview

Installation : How to install MuFastRom

Synopsis : The command line options and tool types

History : What happened before

© THOR-Software

Thomas Richter

Rühmkorffstraße 10A

12209 Berlin

Germany

EMail: thor@einstein.math.tu-berlin.de

WWW: http://www.math.tu-berlin.de/~thor/thor/index.html

1.2 The THOR-Software Licence

The THOR-Software Licence (v2, 24th June 1998)

This License applies to the computer programs known as "MuFastZero" and the "MuFastZero.guide". The "Program", below, refers to such program. The "Archive" refers to the package of distribution, as prepared by the author of the Program, Thomas Richter. Each licensee is addressed as "you".

The Program and the data in the archive are freely distributable under the restrictions stated below, but are also Copyright (c) Thomas Richter.

Distribution of the Program, the Archive and the data in the Archive by a commercial organization without written permission from the author to any third party is prohibited if any payment is made in connection with such distribution, whether directly (as in payment for a copy of the Program) or indirectly (as in payment for some service related to the Program, or payment for some product or service that includes a copy of the Program "without charge"; these are only examples, and not an exhaustive enumeration of prohibited activities).

However, the following methods of distribution involving payment shall not in and of themselves be a violation of this restriction:

(i) Posting the Program on a public access information storage and retrieval service for which a fee is received for retrieving information (such as an on-line service), provided that the fee is not content-dependent (i.e., the fee would be the same for retrieving the same volume of information consisting of random data).

(ii) Distributing the Program on a CD-ROM, provided that

a) the Archive is reproduced entirely and verbatim on such CD-ROM, including especially this licence agreement;

b) the CD-ROM is made available to the public for a nominal fee only,

c) a copy of the CD is made available to the author for free except for shipment costs, and

d) provided further that all information on such CD-ROM is redistributable for non-commercial purposes without charge.

Redistribution of a modified version of the Archive, the Program or the contents of the Archive is prohibited in any way, by any organization, regardless whether commercial or non-commercial. Everything must be kept together, in original and unmodified form.

Limitations.

THE PROGRAM IS PROVIDED TO YOU "AS IS", WITHOUT WARRANTY. THERE IS NO WARRANTY FOR THE PROGRAM, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IF YOU DO NOT ACCEPT THIS LICENCE, YOU MUST DELETE THE PROGRAM, THE ARCHIVE AND ALL DATA OF THIS ARCHIVE FROM YOUR STORAGE SYSTEM. YOU ACCEPT THIS LICENCE BY USING OR REDISTRIBUTING THE PROGRAM.

Thomas Richter

1.3 What's the MMU.library?

All "modern" Amiga computers come with a special hardware component called the "MMU" for short, "Memory Management Unit". The MMU is a very powerful piece of hardware that can be seen as a translator between the CPU that carries out the actual calculation, and the surrouding hardware: Memory and IO devices. Each external access of the CPU is filtered by the MMU, checked whether the memory region is available, write protected, can be hold in the CPU internal cache and more. The MMU can be told to translate the addresses as seen from the CPU to different addresses, hence it can be used to "remap" parts of the memory without actually touching the memory itself.

A series of programs is available that make use of the MMU: First of all, it's needed by the operating system to tell the CPU not to hold "chip memory", used by the Amiga custom chips, in its cache; second, several tools remap the Kickstart ROM to faster 32Bit RAM by using the MMU to translate the ROM addresses - as seen from the CPU - to the RAM addresses where the image of the ROM is kept. Third, a number of debugging tools make use of it to detect accesses to physically unavailable memory regions, and hence to find bugs in programs; amongst them is the "Enforcer" by Michael Sinz. Fourth, the MMU can be used to create the illusion of "almost infinite memory", with so-called "virtual memory systems". Last but not least, a number of miscellaneous applications have been found for the MMU as well, for example for display drivers of emulators.

Unfortunately, the Amiga Os does not provide ANY interface to the MMU, everything boils down to hardware hacking and every program hacks the MMU table as it wishes. Needless to say this prevents program A to work nicely together with program B, Enforcer with FastROM or VMM, and other combinations have been impossible up to now.

THIS HAS TO CHANGE! There has to be a documented interface to the MMU that makes accesses transparent, easy and compatible. This is the goal of the "mmu.library". In one word, COMPATIBILITY.

Unfortunately, old programs won't use this library automatically, so things have to be rewritten. The "MuTools" are a collection of programs that take over the job of older applications that hit the hardware directly. The result are programs that operate hardware independent, without any CPU or MMU specific parts, no matter what kind of MMU is available, and programs that nicely co-exist with each other.

I hope other program authors choose to make use of the library in the future and provide powerful tools without the compatibility headache. The MuTools are just a tiny start, more has to follow.

1.4 What's the job of MuFastZero?

MuFastZero is a mmu.library compatible autovector remapper. These vectors are read frequently by the CPU, but are by default placed in the rather slow chip memory. While it is possible to remap these vectors to faster memory without making use of the MMU, it is more compatible to leave the autovectors at its original place and use the MMU to mirror them to faster memory. This might help - IMHO broken - software to run fine even with the faster system.

The MuFastZero program can also be used to remap the most important system library to fast memory and hence to speedup your system even more. This requires, though, running an additional tiny program "MuMove4K". Hence, MuFastZero is able to replace certain functions of the Oxypatcher.

1.5 Installation of MuFastZero

Installation is pretty simple:

- First, install the "mmu.library": Copy this library to your LIBS: drawer if you haven't installed it yet. It's contained in this archive.

- Copy "MuFastZero" wherever you want.

- In case your system doesn't have any autoconfigurable fast mem and you want to remap parts of the Os from chip ram to fast, you'd keep "MuMove4K" as well.

- Remove all other zero page remappers from your startup-sequence and add the following line:

MuFastZero ON

to enable remapping.

- If you want to use MuFastZero's "FastExec" option to remap even more parts of the system, "MuMove4K" must be installed and run in the startup-sequence as well. The best place for this program is *IN FRONT OF* the SetPatch command. It doesn't take any arguments.

Be warned! Due to the way how the "FastExec" option works, this will make your system incompatible to the ShapeShifter and to programs that hook into the exec resident list after installation of MuFastZero. To be able to run the ShapeShifter, you should remove the "MuMove4K" program by running "NoMuMove4K" before, and you should further more install reset-proof programs BEFORE you run MuFastZero with the "FastExec" option.

That's all.

1.6 Command line options and tooltypes

MuFastZero can be started either from the workbench or from the shell. In the first case, it reads its arguments from the "tooltypes" of its icon; you may alter these settings by selecting the "MuFastZero" icon and choosing "Information..." from the workbench "Icon" menu. In the second case, the arguments are taken from the command line. No matter how the program is run, the arguments are identically.

MuFastZero ON=FASTZERO/S,OFF=NOFASTZERO/S,FASTEXEC/S:

ON=FASTZERO

A simple switch. If present, remapping is enabled. MuFastZero will complain if it is already installed, or the zero page is already remapped or made inaccessible by MuForce.

OFF=NOFASTZERO

Another switch. If given, remapping is disabled again and the mirror image is released.

FASTEXEC

Another switch. If this switch is set, MuFastZero will try to remap parts of the system libraries, most likely the "exec.library" and the "expansion.library" to fast memory. This is only required if you own a board without autoconfiguring memory, hence with "exec.library" in chip mem. To enable this command line option, you *MUST* have run the "MuMove4K" program in the startup sequence.

WARNING: This option is not without its quirks! It will, for example, de-activate all reset-proof programs installed after MuFastZero was run. This is because these programs will obviously modify the remapped copy of execbase and will leave the original execbase unmodified. Since a reset will de-activate the remapping, the modifications will be never seen. Install reset-proof programs before running MuFastZero, and everything will be fine.

Furthermore, remapping the low-memory area or running MuMove4K will make your system incompatible to the ShapeShifter. This is neither the fault of the MuMove4K program, nor of the ShapeShifter, but due to the way how the MacOs is constructed.

When started from the workbench, MuFastZero knows one additional tooltype, namely:

WINDOW=<path>

where <path> is a file name path where the program should print its output. This should be a console window specification, i.e. something like

CON:0/0/640/100/MuFastZero

This argument defaults to NIL:, i.e. all output will be thrown away.

1.7 History

Release 40.2:

This is the first official release. "MuFastZero" does what it is supposed to do, even though the "FASTEXEC" keyword does not yet work. This will be fixed as soon as the "mmu.library" reaches a final state.

Release 40.3:

Fixed the FASTEXEC keyword, added the MuMove4K program to allow remapping of the lower memory parts.