FastExec

COLLABORATORS			
	TITLE : FastExec		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 20, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

FastExec

1.1 FastExec.guide

```
============================================================
                        May 21, 1997

                    Torbjörn Andersson
                        presents
                      FastExec 2.6

                      Public Domain
============================================================
```

```
1. Contents
------------------------------------------------------------
```


1.2 introduction

```
2. Introduction
------------------------------------------------------------
Most things automatically utilize the fast memory in my
Amiga, but not exec.library, the heaviest used library of
the AmigaOS, nor expansion.library. They remain in the slow
chip memory. Since that is a waste of both time and chip
```

memory, I made FastExec.

FastExec will install a resident tag with the priority of expansion.library plus one. After Exec is initialized upon reboot, FastExec's tag will be run. First it will set the cache if it was specified on the command line. Then it patches AddMemList() in exec.library. If specified it will finally add non-autoconfig memory. Then we leave the control back to the system.

When someone calls AddMemList(), the patch uses the original function to add the memory to the system's list. Then it tries allocate fast memory and move exec.library to it.

If you really want to, expansion.library can also be moved to fast memory. AddLibrary() will then be patched to achieve this, similarly to the AddMemList()-patch.

The AddMemList()-patch optionally also relocates the small chip memory list header to fast memory, speeding up memory allocations a little, and the same with memory for the interrupt servers.

More as an experiment, some often used functions in Exec can be patched. The AddMemList()-patch takes care of this too.

Under kickstart 1.2/1.3, MakeLibrary() is automatically patched so library bases will be longword aligned.

1.3 requirements

3. Requirements

FastExec requires true fast memory.

1.4 installation

4. Installation

Edit Startup-Sequence to start FastExec early. Try putting it first, even before SetPatch, if it doesn't work when started later.

If you use the REBOOT parameter, FastExec will reboot the machine and move exec.library to fast memory.

1.5 usage

5. Usage

```
Usage: FastExec [SYSINFO] [REBOOT] [NOEXEC] [LOCAL] [PATCH]
                [FASTSSP] [FASTVBR] [FASTEXP] [FASTMEM] [FASTINT]
                [CACHE 0xhhhhhhhh]
                [ADDMEM <base size attr pri> ...]
```

Any value can be written in both normal and hexadecimal form. Write "0x" in front for hex.

The options has no effect if FastExec already is resident.

SYSINFO

Shows various system information.

REBOOT

Installs resident tag and reboots. Otherwise only the tag will be installed and no reboot will be performed.

NOEXEC

Won't try to move exec.library to fast memory, in case it don't work and you would like to use the other options.

LOCAL

Fast memory will be added with the flag MEMF_LOCAL. This way the RAD-disk will use fast memory.

PATCH

(Ignored under kickstart 1.2/1.3.)

For fun I optimized some functions in exec.library. With this option the lowest level of interrupt handling will be patched, together with the following functions:

- Supervisor()
- ExitIntr()
- Schedule()
- Switch()
- Dispatch()
- Exception()
- Forbid()
- Cause()
- Wait()
- PutMsg()
- ReplyMsg()

I did some speed tests with multitasking not disabled, and I must admit that you usually *don't* notice any difference with this patch. But when I also had a program in the background that simply did a busy loop, then there was a big speed increase of upto 100%.

The patches are mostly simple assembler optimizations. I have used and/or-instructions (sometimes other ones) instead of bset/bclr/btst where possible. ExecBase is embedded in move-instructions instead of reading ExecBase from location $4, which can be useful if you have AGA chipset and/or use Enforcer. Other optimizations are not much to say about.

I have just made optimizations without actually knowing exactly what all the functions do, so beware. :-) There are two versions of the Switch()-function, I haven't tested the one used if you have an FPU.

FASTSSP

Relocates the supervisor stack to fast memory.

FASTVBR

Relocates the vector base register to fast memory.

The 68000 processor has no VBR. (Output from SYSINFO will always say that it is zero).

FASTEXP

Relocates expansion.library to fast memory.

FASTMEM

Relocates memory list headers to fast memory.

FASTINT

Relocates memory for interrupt servers to fast memory.

CACHE

Calls CacheControl() with the supplied parameter. Below are useful values as defined in exec.i. Logically "or" these together. Note that 68000 and 68010 has no cache. 68020 has instruction cache only.

```
CACRF_EnableI       = 0x00000001 (Enable instruction cache)
CACRF_IBE           = 0x00000010 (Enable instruction burst)
CACRF_EnableD       = 0x00000100 (Enable data cache)
CACRF_DBE           = 0x00001000 (Enable data burst)
CACRF_CopyBack      = 0x80000000
```

ADDMEM

Adds memory to the system's memory list. Can be useful if you have non-autoconfig memory. Use "5" for attributes if you don't know what else to use.

I'll take my memory as an example. This is the info I get about it from FastExec with the SYSINFO parameter:

```
MEMORY HEADERS:
Address   Name                 Lower     Upper     Type   Pri
$08000000 DKB1240_Memory       $08000020 $08800000 $0005   10

If it was configured manually and I wanted FastExec to do it
from now on, I would write:

FastExec ADDMEM 0x08000000 0x00800000 5 10
```

1.6 comments

```
6. Comments
------------------------------------------------------------
FastExec stays resident in memory, but after a reset it can
happen that the fast memory is not located at its place.
Then the system won't find ExecBase with its information
about resident things in memory. FastExec and for example
the RAD-disk can't survive a reboot then. I guess this
depends on the memory board one have.
```

1.7 history

```
7. History
------------------------------------------------------------
1.0 (26.11.95)
· First release.

1.1 (14.1.96)
· lib_NegSize under KS 1.3 is now set to correct value and
  not 0.
· Doesn't set the memory attribute flags
  MEMF_LOCAL/MEMF_24BITDMA under KS 1.3.
· The name string "expansion ram" is used for the fast
  memory under KS 1.3, instead of "Fast Memory".
· The priority that the fast memory has will not be changed,
  it was always set to 10 before.
· Could get wrong idea of where the kickstart was located,
  some checking together with alternative methods makes it
  safer now.
· If kickstart version is 2.0 or higher, FastExec will
  search through the kickstart for everything it needs from
  it. If FastExec will work doesn't depend on the exact
  kickstart version, but more on how it is built. If there
  aren't too big differences, FastExec should now work with
  any kickstart version.
· Changed some output from the SYSINFO argument, and
  documented that the argument exists. :)
· BOARDADDR handles more than one address.
```

```
2.0 (21.4.96)
· Rewrote the main code, should work better out there.
· Can relocate a few extra things to fast memory, relocating
  expansion.library is optional.
· Added patches for much used functions in exec.
· Can set caches very early on bootup.
· Can add non-autoconfig memory.
· SYSINFO shows all libraries, and in what type of memory it
  is located.
· FastExec is now freeware.

2.1 (25.8.96)
· Fixed SYSINFO output of FPU.
· When I changed mouse, FastExec always disabled the fast
  memory. I guess I could have fixed that, but I removed
  the option instead.
· Doesn't free the old chip memory header under
  kickstart 1.2/1.3, as it is misaligned (FASTMEM option
  caused FreeTwice-alert).
· Fixed bug with FASTINT under kickstart 1.2/1.3.
·~Added NOFASTEXEC option.

2.2 (29.8.96)
· Added LOCAL option.

2.3 (11.9.96)
· Rename NOFASTEXEC option NOEXEC.
· Added PCMCIA option.
· AddLibrary()-patch restores old function after
  expansion.library has been moved.

2.4 (9.11.96)
·~Gives some errors if installing fails.
· Restores old AddMemList()-function after the first memory
  of type MEMF_FAST has been added.

2.5 (18.1.97)
· Made FastExec public domain, and included the source code
  in the archive.

2.6 (21.5.97)
· Fixed bug in Switch()-patch for machines with FPU.
· Sets CACRF_WriteAllocate when CACHE option is used.
· Allocates memory for resident tag with MEMF_REVERSE.
· Uses ReadArgs() (KS 2.0+).
·~Doesn't free memory for old exec.library under KS 1.3 if
  eb_ExecBase in expansion.library can't be replaced.
· Removed PCMCIA option.
```

1.8 credits

```
8. Credits
-----------------------------------------------------------

Thanks to:
```

· All who registered when FastExec was shareware.
· All who have helped me in my more or less successful
  attempts in fixing problems.
· Amiga Shopper for politely asking before putting FastExec
  on their coverdisks and even sending me a copy of the
  magazine.

1.9 author

9. Author

FastExec was made by Torbjörn A. Andersson.

Email:
d95ta@efd.lth.se

Home page:
http://www.efd.lth.se/~d95ta

Home address:
Torbjörn Andersson
Knöppletorp 4379
S-380 31 LÄCKEBY
SWEDEN

I love you - AMIGA