

in

Stefan Fuchs

COLLABORATORS

	TITLE : in		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Stefan Fuchs	August 26, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	in	1
1.1	PatchSetFunc Documentation	1
1.2	Motivation	1
1.3	Features	2
1.4	Requirements	2
1.5	Installation	2
1.6	Notes	3
1.7	Compatibility	3
1.8	Bugreports	3
1.9	History	4
1.10	Distribution	5
1.11	Author	5

Chapter 1

in

1.1 PatchSetFunc Documentation

PatchSetFunc V1.40 -- Redirect SetFunction() to patch.library

Copyright © Stefan Fuchs, August 1996. Freely distributable.

Click on one of the following topics for more detailed information:

Motivation	Why I wrote PatchSetFunc
Features	Why using PatchSetFunc
Requirements	What is needed for using PatchSetFunc
Installation	How to install PatchSetFunc
Notes	What you should know as well
Compatibility	What programmes do not work
Bugreports	What I need, if you find a bug
Distribution	Conditions for redistribution
History	Information about past and present versions
Author	How to get in touch with the author

Click on the CONTENTS button at the top to return here at any time.

1.2 Motivation

This one was written as a support program, for my patch.library. It patches the exec.library/SetFunction() to use patch.library instead of the normal OS-function.

Although I'm aware that there exist a couple of programs, which perform the same task quite well (e.g.: SetManager by Nicola Salmoria, SaferPatches by Martin Adrian, PatchControl etc..), I thought it would be nice, if more programmes could use my patch.library.

1.3 Features

FEATURES:

If you're not expert with the Amiga operating system, all you need to know about PatchSetFunc is that it will, under certain circumstances, avoid GURUs.

If you're an expert, here come the details. There's a function in exec.library, whose name is SetFunction(). You know that Amiga libraries contain a 'jump table', where the address of every function in the library is stored. Library functions are always executed via the jump table, and NEVER through absolute addressing. SetFunction() allows the user to modify the jump table, in order to insert a custom routine in place of the standard one. For example, one could modify the DisplayBeep() vector to play a sound instead of flashing the screen.

SetFunction() is a very useful feature of the Amiga operating system, but it has the major drawback that there's no system supervising of concurrent use of the same vector by different tasks. Under certain circumstances, that can be deadly. Before removing its custom vector, a program should check that it was not modified again. However, very few do that, and so if you do not terminate such program in REVERSE order of when you executed them, you'll probably get a system crash.

Another fault of Kickstart 1.x is that the dos.library vectors cannot be modified by SetFunction(), because they do not follow the conventions established by Commodore itself.

All this could be fixed if all programmers would use the patch.library.

Unfortunately patch.library is quite new and many programs still use SetFunction().

So I wrote PatchSetFunc, which allows any program to take profit from patch.library.

1.4 Requirements

SYSTEM REQUIREMENTS:

- Any Amiga with any Kickstart version
- patch.library v4+ must be installed in libs:
(make sure you have the latest version)

1.5 Installation

The program 'PatchSetFunc' should be placed as the first command in the startup-sequence after the SetPatch command. You must specify the full path, if you place the command in a directory other than c:.
You may also want to redirect the output to nil:.

The patch.library should be copied to LIBS:.

1.6 Notes

NOTES & WARNINGS & KNOWN BUGS:

- Programs directly reading addresses from the jump-table, may fail to quit! (This behavior is illegal and not supported by Commodore!)
- In order to make patch.library to work with XOper a kludge has been added, which stops PatchSetFunc the exec.library/Switch() function (which in fact does the actual multitasking in the Amiga) of being patched
- Some programs patch the Expunge() function of a library, to ensure that the library won't get expunged, while a patch is active. If the support program PatchSupervisor is active PatchSetFunc ignores patches to the Expunge() function.
- Programmers should use patch.library directly to ensure maximum safety, when patching library functions
- A developer package with full documentation and includes is available on Aminet (util/libs/PatchLibV4.lha) or (util/libs/PatchLib.lha)
- This software will show its full capabilities once the preferences editor for the patch.library is available in the near future. Programs can be excluded from using PatchSetFunc....

1.7 Compatibility

In general PatchSetFunc should work together with all programs, that know about SetManager, because for these programs it will look like SetManager was active. (The public messageport 'SetMan' can be found via FindPort())

Other programs should be configured (if possible), not to check, if a patch was installed after its patch.

Program: Change when using PatchSetFunc:

XOper V2.x turn SmartPatch OFF

The following programs are known not to work fully with patchsetfunc: enforcer

```
Error:    Can not be turned off
Reason:   unknown
Solution: excluded from PatchSetFunc
```

Executive

```
Error:    crashes during install
Reason:   unknown
Solution: excluded from PatchSetFunc
```

NoDraw + NoFill from Aminet CD 11: util/wb/NoFillNoDraw.lha

```
Error:    crashes after install/does not work
Reason:   Direct read from jump-table/depends on undocumented
          values residing on stack
Solution: do not use
```

1.8 Bugreports

If you find any programs not working together with PatchSetFunc please send me a bugreport containing the following information:

- What versions of patch.library and PatchSetFunc are you using?
- What is the name and version number of the program not working with patchsetfunc and what is the archivname on the Aminet (if possible)?
- What steps are to be taken to reproduce the bug?
- What exactly goes wrong (crash, guru,...)?
- What patches are installed on your system?
The output of the PatchList program (Aminet:util/libs/PatchLibV4.lha) would be nice.
- Are there any alerts with the Gurunumber: \$44000xyy ?
These alerts are send out by PatchSetFunc, if patch.library reports an error. There is no way to tell this an application, because the original SetFunction() never fails.
x indicates the function, that failed.
yy is the errorcode as defined in the patch.h include file supplied in the PatchLibV4.lha or util/libs/PatchLib.lha archive

1.9 History

- V1.41 ENHANCEMENT: Sets a private bit in the patch.library base structure, if active ←
- DOCUMENTATION: Problems with NoDraw & NoFill
- V1.40 ENHANCEMENT: Automatically extends the stack to at least 2000 bytes for each patch, if patch.library V5+ is installed
- V1.35 FIX: Patching of Expunge() is now possible for programmes other than unarp (This fixes crashes with GPFax)
- V1.33 FIX: By passing a wrong tag to patch.library, too many internal poject-structures were created (This reduces memory usage)
- V1.32 ENHANCEMENT: Now does nothing, if the librarybase or the function passed to SetFunction() is NULL.
(This fixes problems with the buggy CEDPatch)
(Reported by Robert Hofmann)
- ENHANCEMENT: Docs now in AmigaGuide format
- V1.31 FIX: No longer used for Executive, because of crashes when removing patches
- V1.30 FIX: Calling SetFuction() for the same patch for the same function twice installs this function now only once
(This fixes crashes with PatchSetFunc + RTPatch + Iconian)
(Reported by Alberto Caporro, Ariel Magnum, Niels Bache)
- V1.28 FIX: Loadresource reenabled (crashes were caused by bugs in earlier versions of patch.library, which have been fixed)
- V1.25 FIX: Disable patch after removing failed
- V1.24 FIX: Display some alerts, if something goes wrong
- V1.23 FIX: No longer used for LoadResource, because of crashes
(Reported by Ariel Magnum)
- V1.22 FIX: No longer patches the exec.library private functions Schedule() and ExitIntr(). This caused ShapeShifter to crash on exit.
- V1.21 ENHANCEMENT: Now supports project management of patch.library V4
-

V1.10 ENHANCEMENT: Now uses less stack

V1.06 FIX: The public message port was spelled 'Setman' instead of 'SetMan'

V1.02 RELEASE: First public release

1.10 Distribution

This material is © Copyright 1993-96 by Stefan Fuchs. All rights reserved.

It may be distributed freely as long as the following restrictions are met:

- The distributor may charge a fee to recover distribution costs.
The fee for diskette distribution should not be more than
the cost to obtain the same diskette from Fred Fish.
- The distributor agrees to cease distributing the programs and
data involved if requested to do so by the author.
- You may copy and distribute verbatim copies of the program's
executable code and documentation as you receive it, in any
medium, provided that you conspicuously and appropriately
publish only the original, unmodified program, with all
copyright notices and disclaimers of warranty intact and
including all the accompanying documentation, example files and
anything else that came with the original.
- If you are interested in including any of this material in a commercial
product, you should contact the author for his permission.
- The author will not be liable for any damage arising from the
failure of the programs or the library to perform as described,
or any destruction of other programs using the library residing
on a system. While I know of no damaging errors, the user of this
package uses it at his or her own risk.

This package may be distributed in PD-series (e.g.: the Fred Fish library) or
on the Aminet.

1.11 Author

To contact the author for bugreports, hints, ideas, donations,
write to:

Stefan Fuchs E-Mail: snfuchs@sokrates.franken.de
Oskar-von-Miller-Str. 49 Fido: Stefan Fuchs@2:2490/1901

D-90478 Nuernberg
GERMANY
