

Asm-Pro

COLLABORATORS

	<i>TITLE :</i> Asm-Pro		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		May 28, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Asm-Pro	1
1.1	Asm-Pro V1.11 By Genetic	1
1.2	Installation	1
1.3	Thanks	1
1.4	Contact Addresses	2
1.5	About Asm-Pro.	2
1.6	Disclaimer...	2
1.7	Requirements...	3
1.8	Commandline.	3
1.9	Project	5
1.10	Editor	7
1.11	Memory	7
1.12	Insert	8
1.13	Assembler	9
1.14	Monitor	9
1.15	Diskette	10
1.16	Miscellaneous	11
1.17	Special features.	12
1.18	Special features - %Getdate/%Gettime.	12
1.19	Special features - IncIFF(p).	12
1.20	Asm-Pro - The story.	15
1.21	15

Chapter 1

Asm-Pro

1.1 Asm-Pro V1.11 By Genetic

Asm-Pro V1.11 - Coded by Genetic in 1998

Choose your destiny:

[About Asm-Pro](#)

[Why? What?](#)

[Disclaimer](#)

[Commandline](#)

[Special features](#) (like inciff etc.)

[Installation](#)

[Contact Addresses](#)

[Thanks](#)

1.2 Installation

Installation

- Create an Asm-Pro directory on your favourite drive and copy all files/dirs from disk to this directory.
- Optional you can assign "Asmprom:" to this directory so the Asm-Pro.guide and the Regsdata file can be found.

I am sure you've got the system include files so they are not included. If you don't you should get the Amiga developer CD.

1.3 Thanks

Thanks to

Special thanks go to the Beta testers for their constructive comments, bug reports and ideas.

- One/Genetic MC68060/50 MPC604/200 64MB CyberVision 64
 - Tib/TFA MC68030/50 8MB
 - P-O Yliniemi MC68060/50 MPC604/200 134MB CyberVision 64
 - Thomas Wittwer MC68030/50/fpu 32MB
-

- Peter'ViTAL'Eriksson MC68060/50 36MB CyberVision 64
- Scorpion/Silicon MC68030/50 8MB
- Cliff Earl MC68000

1.4 Contact Addresses

Contact Addresses

Asm-Pro Homepage : <http://surf.to/asmpro>

Or if that fails : <http://home.worldonline.nl/~solognt>

E-Mail

Guide - Feanor : aramaker@concord-eracom.nl

Asmpro - Solo : solognt@worldonline.nl

Surf to Genetic Homepage : <http://surf.to/genetic>

1.5 About Asm-Pro.

About Asm-Pro

Asm-Pro is a MC680x0 marco assembler with integrated editor, debugger, linker and monitor for our Amiga.

With this program you have all the tools you need for the development of new software in assembly.

It has full mc680x0, FPU and MMU support (and ppc support will be added when i get my ppc board).

This version of Asm-Pro still contain a number of (known) bugs, i desided to release it anyway. A program as complex as this will never be bug free.

Feel free to tell me what you think of it and report me any bugs you might find. But first check the known bugs section in the history file and check my [homepage](#) for a newer version of Asm-Pro and/or the Historyfile.

Check out the [why? what?](#) section for more background info.

Solo/GnT

Credz:

All coding of Asm-Pro by Solo / Genetic

Asm-Pro Guide by Feanor / Genetic

1.6 Disclaimer...

Disclaimer

No guarantee of any kind is given that Asm-Pro is 100% reliable. You are using it at your own risk. The author takes no responsibility for any damage which is caused by using this program.

1.7 Requirements...

Requirements

Asm-Pro needs the following configuration of hardware and software to be present:

- Any Amiga equipped with an MC680x0
- Kickstart version 39.106 or above
- ReqTools.Library for Kickstart V37
- MathTrans.Library for Kickstart V39

1.8 Commandline.

Asm-Pro Command summary

The command line is the command centre of Asm-Pro, recognized by the prompt :

>

Next are the commando's which can be used.

Project

ZS - Delete source text

O - Restore deletions of source text

R - Load Source code

RB - Load Binary data

RO - Load Object module

W - Save source text

WB - Save Binary data

WO - Save Object module

WL - Save Linkfile

I - Insert

U - Update file

ZF - Remove file

WP - Save Preferences

=M - Enlarge working memory

! - Quit Asm-Pro

!! - Quit Asm-Pro Fast

!R - Restart Asm-Pro

=C - Change colors

Editor

T[line] - Start of source (or line n)

B - End of source

L[Text] - Text search

ZL[line] - Delete lines from cursor position

P [line] - Print lines from cursor position

EL - Expand labels with....

Memory

M[.amount][address] - Memory edit

d[address] - Disassemble

H[.amount][address] - View memory, hexadecimal

N[address] - View memory, ASCII text

S[.amount] - Search

F[.amount] - Fill

C[.amount] - Copy

Q - Compare

Insert

ID - Insert Disassembled code

IH[.amount] - Insert Hexadecimal code

IN - Insert ASCII code

IB - Insert Binary code

IS - Insert Sinus

Assembler

A - Assemble in Editor

AO - Assemble with optimisation

AD - Assemble with debug information

=S - View Symbol table

Monitor

J[address] - Jump to subroutine (JSR)

G[address] - Jump immediately (JMP)

K[step] - One step mode

X[register] - View and/or change register

ZB - Remove breakpoints

Diskette

RS[drive] - Read Sector

RT[drive] - Read Track

WS[drive] - Write Sector

WT[drive] - Write Track

CC[drive] - Calculate Bootblock-Checksum

W - Load External data

Miscellaneous

CS - Create Sinus in memory

Y [command] - Execute Dos command

V [path] - Show contents of path

> - Revert output to PRT: or Dfx:

?[expression] - Calculate number expression

return - Empties screen

=R - Custom chip information

PS - Parameter Set

1.9 Project

PROJECT

ZS - Zap Source

Deletes the program text, the copybuffer (clipboard) and the code.

The program text can be undeleted with the "o" (old source) command but only if there is no new source text entered in the editor window.

O - Old Source

With this command you can retrieve the source text after a zs command.

R - Read

Reads a source text into the editor. Asm-Pro assumes it is a ASCII text.

If you want to alter the normal '.s' extension change "PROJECT/PREFERENCES/SOURCES.S".

RB - Read Binary

This command will load binary data into a memory address.

After the file name the start and end address will be asked.

BEG>

END>

BEG (Begin) is the first address that will be filled with data, END (End) the last.

BEG>\$70000

END>\$71000

Loads the first \$1000(=4096 bytes) from a binary file into the memory, starting at address \$70000. To load the complete file leave the END> empty.

Example :

BEG>\$70000

END>

RO - Read Object

Reads an executable into memory. It is sometimes possible to execute it.

Some programmes require input from the CLI. To get this right you must fill in the registers A (pointer to parameters) and D (Length) yourself or use the PS command.

When loading an empty executable (" ") all memory is cleared.

W - Write

Writes the source code to a file. The file is a normal ASCII file so editing with other text editors is also possible.

Usually the .S extension will be placed behind the name. If you want a different extension switch of the "PROJECT/PREFERENCES/SOURCE.S". Now you can choose your own extension with the ![extension] command. Example:

```
>!.asm
```

WB - Write binary

Writes a binary file to disk. You have to give the begin and end address.

Example:

```
BEG>$70000
```

```
END>$71000
```

Saves \$1000(=4096 bytes) from address \$70000 as a binary file.

WO - Write Object

After assembling it is possible with the WO command to write the object/executable to disk.

Use WL to write a Linkfile.

I - Insert

With this command it is possible to insert a text file into the existing source text.

U - Update File

Writes the source text to disk, updating the file. Same as W (write).

ZF - Zap File

Deletes a file from disk.

Zi - Zap Include memory

Deletes include files from memory. The include files are kept in memory to speed up the assembling process.

Wp - Write Preferences

Creates or updates the Asm-Pro.Pref file.

=M - Expand Memory

This command makes it possible to expand the working memory (as long as there is memory available) without losing anything.

! - Quit Asm-Pro

Quits Asm-Pro.

!! - Quit Asm-Pro Fast!

Quits Asm-Pro without exit or restart requester and changes are not saved.

!R - Restart Asm-Pro

Restarts Asm-Pro. Memory is deallocated and startup window will be shown even when disabled. Changes are not saved.

=C - Change Colors

To override the standard preferences colors you can use this command.

A window will appear making it possible to adjust the colors.

1.10 Editor

EDITOR

T- / Top of file

Jumps to the line number specified, otherwise jumps to the start of the source.

>T100 : Jumps to line 100

>T : Jumps to the start of the source

>t-1 : Jumps to the end of the source

L- Locate text

Searches for the string specified.

>Lmove : Searches for all move operands in the source.

To search again simply type : >L

NOTE : Between the L commando and the string cannot be a space.

ZL- Zap Lines

Deletes all lines from actual cursor position.

>ZL100 : deletes 100 lines from actual cursor position.

>ZL-1 : deletes all lines from actual cursor position.

P- Print Lines

Prints from cursor position the amount of lines specified.

To print the lines select "Project/Preferences/Printerdump" or press

Ctrl-P.

>P100 : Prints 100 lines from cursor position

>P-1 : Prints all lines from cursor position

EL- Extend Lables

If more than one programmer is working on the same source, there is a change that lable names are used more than once. To avoid this the EL commando is usefull

Asm-Pro will ask with what you want to extend the lables:

Extend Lables with>

1.11 Memory

MEMORY

M- Memory Edit

Gives direct edit possibility in memory. You can enter text or hexadecimal values.

D,@D- DisAssemble

Goes to the assemble modus, the memory is disassembled. You can scroll, jump and edit. To undo changes press Escape

With @D you only get 12 lines on your screen, so half-screen.

H,@H- HexDump

Goes to the hexdump modus. Again you can scroll, jump and edit. If you want to edit longwords the H.L commando is more interesting.

With this you can edit complete longwords instead of just bytes.

With @H you only get 12 lines on your screen, so half-screen.

N@N - AsciiDump

Goes to the ASCII modus. Again you can scroll, jump and edit. The view is in rows of 64 letters and again you only get 12 lines on your screen, so half-screen.

@A - Assemble mem

Assembles directly into memory.

@B - BinaryDump

Displayed is a view with 8 lines of binary data.

S - Search in memory

Example:

>S

Beg>\$10000

END>\$20000

After entering the start and end address you can enter the data you want to search

DATA>123 /or 4321.I /or "hello" /or \$4532.w /or %101001.b

Default search size is byte.

F - Fill Memory

Be carefull with this commando. If you fill the wrong part of the memory you can crash the program or even your computer.

C- Copy Memory

Copies a part of the memory to an other part of the memory.

Q- Compare Memory

Compares two parts of the memory. When not equal the address of the first not equal byte is returned.

1.12 Insert

INSERT

ID - Insert DisAssemble

This is a very powerful commando. It gives you the ability to disassemble memory and place it directly into the source code.

The new code will be supplied with lables (if possible), so it is ready for use.

The one disadvantage is that the commando is rather slow. For the real disassemble work you can better use Resource or some other proper program.

IH - Insert Hexdump

Gives you the possibility to add hexadecimal text into your source code.

IN- Insert ASCII

Gives you the possibility to add text as DC.B strings into your source code.

It will automatically use hexadecimals if not defined as ASCII.

IB - Insert Binary

Gives you the possibility to add binary as DC.B into your source code.

IS - Insert Sinus

Gives you the possibility to add a Sinus into your source code.

1.13 Assembler

ASSEMBLER

A- Assemble

This will start the assembling, same as Amiga-Shift-A.

@A - Assemble to memory

AO - Assemble optimized

Normally assembles the source but will try to optimize the branch distance to the shorter .S version. When the branch is more than -128 or +128 away, the branch is changed into a .W version.

AD - Assemble debug

Assembles the source and then jumps to the debugger.

=S - Symbol table print

Shows all global variables after assembling. These variables can also be printed, set "PROJECT/PREFERENCES/PRINTER DUMP" and select Listfile.

1.14 Monitor

MONITOR

J- Jump to address

Jumps to the specified address like a subroutine (JSR).

When no address is specified the jump will be to the start of the source.

Example :

>JStart

Jumps to the label "Start".

G- Go to address

Same as the J command with the addition that it is possible to set a breakpoint or stop at an illegal command.

K- Single step, n steps

Steps for n steps through the source from actual program counter.

See Debugger for more information.

X- Change - edit registers

Can be used in two ways:

>X : view all registers

>XD2 : view one register (expl D2)

Just typing X will show all registers, including USP, SSP, SR and PC.

Flags of the statusregister will be shown as characters.

All changes in the registers since last view will be shown invers colored.

1.15 Diskette

DISKETTE

RS - Read Sector

Read sectors from a disk. An Amiga disk has 80 tracks on each side which are seperated into 11 sectors. This makes $80 * 2 * 11 = 1760$ sectors, each sector is 512 bytes large. The bootblock consist of 2 sectors, starting at sector 0.

The naame of the disk is in the rootblock which can be found on track 80.

Example :

To read the bootblock of DF1: into address \$70000

>RS1 : diskdrive 1

RAM PTR>\$70000 : Pointer to RAM-address

DISK PTR>0 : Sector number

LENGTH>2 : Amount of sectors to read

RT - Read Track

See RS

WS - Write Sector

See RS

WT - Write Track

See RS

CC- Calculate checksum

The first thing loaded by the system is the bootblock.

With this information it determinse what kind of disk it is, KICK=Kickstart,

DOS=dos, BAD=unknow format.

The bootblock can also contain a executable. To check if the data is usable, the systems checks the chekcksum. With CC this checksum can be recalculated.

This can be useful when programming your own bootblock or in case of a virus.

Example:

>CC1

Will calculate the checksum on the bootblock in DF1:

When a RS1 or RT1 commando is issued, the checksum will automatic be calculated.

E - Extern files load

Data which is pointed at with the external directieve, will be loaded when the E commando is given. Specify a numner if you do not want to load all external data.

1.16 Miscelaneous

MISCELANEOUS

CS- Create sinus in memory

This commando creates a Sinus.

Example :

DEST> : Place in memory where sinus must be placed

BEG> : Starting Angle

END> : Ending Angle

AMOUNT> : number of entries in sinus table

AMPLITUDE> : Highest/Lowest value of the sinus

YOFFSET> : The middle of the sinus

SIZE> : B(ytes)/ W(ords) / L(ongwords)

MULTIPLIER> : Afterwards number multiplier

HCORRECTION> : Halve step correction

RCORRECTIE> : ROUND in stead of INT (0,7=1 in stead of 0)

Y - Execute dos commando

With this commando you can start a Dos program while working in Asm-Pro

Example :

>Ydiskmaster

V - view directory

Shows the contents of the path specified. Necessary when the Req.Library is shut down.

Example

>VDF0:

Shows the contents of DF0:

Free space is in bytes.

> - Specify output

Reverts the output of a program to disk or printer. If the output is to disk the file will get the .TXT extension. To stop the output repeat the commando without a path.

? - Calculate value

Usage of all operators and defined lables is possible. The result is shown in hexadecimal, decimal, ASCII and binary.

R - register information

Has the following options :

>=R <002> value search with address and register.

>=R DMACONR name, search with the original name.

>=R gives a list of all registers

PS - Parameter Set

Sets the parameters used in programs which have a CLI input.

Asm-Pro calculates the length and puts it into the correct register.

1.17 Special features.

Special features.

IncIFF(p)

%Getdate/%Gettime

1.18 Special features - %Getdate/%Gettime.

Special features - %Getdate/%Gettime.

Implemented %getdate and %gettime these commands will generate the current date or timestring. this is usefull for a version string or about window.

Im working on support for different date formats though an extra option but this one will be default (dos date format). (P-O ;-)

works like this:

versionstring:

dc.b "\$VER: Asm-Pro v1.03e ("

%getdate

dc.b ") By Solo/Genetic.",0

This will generate the following version string:

"Asm-Pro V1.03e (05-Apr-98) By Solo/Genetic."

Same goes for %gettime (ex. "21:06:09").

1.19 Special features - IncIFF(p).

Special features - IncIFF(p).

=====

Example source IncIFF(p).

- INCIFF

Include iff picture as raw-blit or raw normal data with or without colormap

Syntax:

Picture: INCIFF {filename}[,conversion mode[,Cmap placing[,Cmap mode]]]

{filename} = The name of the IFF ILBM file to include.

{conversion mode} = "RN" or "RB" for RAW-NORMAL or RAW-BLIT format

{Cmap placing} = "A","B","N" for Cmap After,Before or None

{Cmap mode} = "ECS","AGA" for word or longword Cmap.

If no options are given Asm-Pro will convert to defaults : RAW-BLIT

No Cmap

Word Cmap

The Cmap in ECS mode will be decoded to WORDS.

The Cmap in AGA mode will be decoded to LONGWORDS.

ECS Cmap Example:

DC.W \$0000,\$0F00,\$00F0,\$000F ; COLOR00 - COLOR03

AGA Cmap Example:

DC.W \$0000,\$0000 ; COLOR00 High-RGB,Low-RGB

DC.W \$0F00,\$0F00 ; COLOR01 High-RGB,Low-RGB

DC.W \$00F0,\$00F0 ; COLOR02 High-RGB,Low-RGB

DC.W \$000F,\$000F ; COLOR03 High-RGB,Low-RGB

- INCIFFP ((copper)palet picture)

Include pallet from iff picture in your program.

Syntax:

CopperColor: INCIFFP {filename}[,Coppermode[,Bankoffset][,coloroffset]

colorlist: INCIFFP {filename}[,list type]

{filename} = The name of the IFF file containing the CMAP hunk.

{copper mode} = "CE" or "CA" for ECS or AGA copper list

Default is "CE"

{Bankoffset} = [0-7] offset for start bank. (CA only)

Default is "0"

{coloroffset} = [\$180-\$1be] color offset inside a bank

Default is "\$180"

{list type} = "12" or "24" for 12 bits or 24 bits color list

Default is "12"

If no options are given Asm-Pro will convert to default : colorlist 12bit

colorlist:

a inciffp adventure/pics/hall.ehb

b inciffp adventure/pics/hall.ehb,12

c inciffp adventure/pics/hall.ehb,24

coppercolors:

d inciffp adventure/pics/hall.ehb,CE

e inciffp adventure/pics/hall.ehb,CE,\$190

f inciffp adventure/pics/hall.ehb,CA

g inciffp adventure/pics/hall.ehb,CA,\$180

h inciffp adventure/pics/hall.ehb,CA,2

i inciffp adventure/pics/hall.ehb,CA,2,\$180

12BitsColorlist:

dc.w \$0000

dc.w \$0F00

dc.w \$00F0

dc.w \$000F

24BitsColorlist:

dc.l \$00000000

dc.l \$00FF0000

dc.l \$0000FF00

dc.l \$000000FF

ECScoppercolorlist:

dc.w \$0180,\$0000

dc.w \$0182,\$0f00

dc.w \$0184,\$00f0

dc.w \$0186,\$000f

AGAcoppercolorlist:

dc.w \$0106,\$0C40

dc.w \$0180,\$0000,\$0182,\$0EEE

dc.w \$0184,\$0100,\$0186,\$0110

dc.w \$0188,\$0111,\$018A,\$0220

dc.w \$018C,\$0221,\$018E,\$0320

dc.w \$0106,\$2C40

dc.w \$0180,\$0981,\$0182,\$0777

dc.w \$0184,\$0982,\$0186,\$0984

dc.w \$0188,\$0A91,\$018A,\$0995

dc.w \$018C,\$0996,\$018E,\$0A92

1.20 Asm-Pro - The story.

Asm-Pro, the story.

Asm-Pro, the story...

Asm-Pro is based on the great Asm-one by Tfa (v1.28). I started this project after I used trashm'one (with it's nice inciff) for a while but it doesn't understand 68020+ code, so I started resourcing trashm to put some 68020 commands in it. After a while I thought it might be easier to put inciff and some other functions in asm-one. So I started 2 resource asm-one (several times) and here we are.

There are a lot of changes since asm-one v1.28. The biggest change is that I rewrote the entire editor and all other character output stuff to be system friendly so it also works on your favourite gfx-board. For a list of new features and updates see the major changes and for a more detailed description I suggest you read the history file.

Major Changes:

- Editor and other text output is system friendly now and works great on your favourite gfx-card.
- Redesign of the Asm-Pro look.
- Included a font requester and made the editor font sensitive.
- New gui for starting up Asm-Pro, workspace mem type and size can be saved with wp (write prefs).
- Multiple source file extensions (try ".sl.asml.i") as source extension to get all sources and includes in the file requester.)
- Debug registers have their own window now.
- Stack size can be changed from shell or icon so even very big sources don't crash your system.
- Added the new mc68060 commands.
- Changed block insert from Amiga-i to Amiga-v (amiga-i will still work though).
- Included my own IncIFFp routines, so it works now.
- Syntax coloring added (comment only for now).
- Added %getdate and %gettime it returns system time and date in ascii.
- Set mark keys should now also work with other keymaps (like swedish or german) So pressing Amiga+Shift+2 will set Mark 2 etc. same goes for delete conditions in debugmode.

Mapped AGA guide from Amiga- to HELP-key so it won't get in the way.

- And some more minor changes and fixes.

1.21

Leeg.