

routines_lib

COLLABORATORS

	<i>TITLE :</i> routines_lib		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 23, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	routines_lib	1
1.1	routines_lib.guide	1

Chapter 1

routines_lib

1.1 routines_lib.guide

R O U T I N E F I L E S

Copyright © 1993-1996 by Stefan Walter, Rene Eberhard, and Daniel Weber

Introduction

Routine files are a library of useful routines that provide a simple and time saving method for assembly programming. Special macros have been designed that ensure that only the called routines get assembled. This makes the routine files a good foundation for assembly programming.

All the routine files are part of the ProAsm Software Package.

The routine files can be freely copied and distributed, provided that:

- 1) No fee is charged for such copying and distribution, and
- 2) They are distributed ONLY in their original, unmodified state.
- 3) This AmigaGuide file must be included to the distribution.

Disclaimer

The authors cannot be held liable for the suitability or accuracy of this manual and the routine files it describes.

Any damage directly or indirectly caused by the use or misuse of this manual and the routine files it describes is the sole responsibility of the user her/himself.

The Routine Files

Unfortunately there exists currently no complete documentation for these routine files. Please refer to the files themselves for further information.

- alert.r - Alert support routines.
 - amigaguide.r - AmigaGuide support routines.
 - amigaguideasync.r - Asynchronous AmigaGuide support.
 - AppIcon.r - Routines for Workbench 2.0's appicon support.
 - ARexx.r - For ARexx support.
 - ASLSupport.r - For support of the ASL file requester.
 - basicmac.r - Macros for selective routines assembly.
 - commodity.r - Contains routines for commodity support.
 - configfile.r - Configuration file support routines.
 - conio.r - Routines for console window input/output.
 - conoc.r - Single console window open/close routines.
 - conread.r - Read routines for the console window.
 - conreadpkt.r - Read routines for the console window using packets.
 - CRC16.r - Routines to calculate an Ansi CRC16 checksum.
 - DiskObjectSupport.r - DiskObject support routines.
 - dosfile.r - Contains DOS file handling routines.
 - doslib.r - DOS library open and close routines.
 - easylibrary.r - Routines to open and close libraries simple.
 - extmsg.r - External message support routines.
 - GadgetGroupSupport.r - GadgetGroup support routines for use with GTFace.
 - graphicslib.r - Graphics library open and close routines.
 - gtface.r - Window handling and gadtools interface routines.
 - gtfdefs.r - Definitions for GTFace.
 - gtfguido.r - GUI macros for GTFace.
 - gtfmacros.r - Macros for GTFace (gadgets and menus).
 - gtfsupport.r - Various GTFace support routines.
 - gtfsupport_reb.r - More GTFace support routines
 - gtfxdefs.r - External definitions for GTFace.
 - intuitionlib.r - Intuition open and close routines.
 - IntuitionSupport.r - Routines for the intuition BusyPointer.
 - locale.r - Support routines for locale (locale.library).
 - locks.r - Routines for locks, files, and directories.
 - memory.r - Memory handling routines.
 - numbers.r - Various routines for number conversion.
 - numbers.mac - String to number macros.
 - packets.r - DOS packet handling routines.
 - paraliner.r - UNIX like parameter line parser.
 - parse.r - Routines for text parsing.
 - patch.r - Library function patch routines.
 - ports.r - Contains ports, signal, and message handling routines.
 - progressbars.r - Routines to handle progress bars easily.
 - qsort.r - Quicksort sorting algorithm.
 - readargs.r - Interface routines to ReadArgs().
 - readrex.r - Routines for a passive ARexx port.
 - requester.r - ASL and REQ file requester routine.
 - reset.r - Software reboot routine.
 - script.r - Routines to deal with batches.
 - scrollbar.r - GTFace appendum for horiz./vert. scrollbars.
 - shortcut.r - Routine to wait for a shortcut.
 - startup4.r - Enhanced CLI and Workbench startup code (with detach).
 - string.r - String support routines.
 - stringmacros.r - String support macros.
 - structs.r - Macros for various structures.
 - support.mac - Support macros.
-

- tasks.r - Some little routines for tasks.
- tasktricks.r - More routines for tasks.
- tooltypes.r - Routines to get ToolTypes from a Workbench started program.
- ToolTypeSupport.r - ToolType support routines.
- TypeOfProcessor.r - Contains routine to determine installed processor and coprocessors. ←

How Do I Use Them?

The usage of the routine files is very simple when you have understood once the underlying scheme. Look at the following example:

```

;
; A small example using the 'dosfile.r' routine file
;
incdir "Work:asm/routines" ; the path where the files are placed

;
; the following line includes the basic macro package that is used to
; handle the single routines or structures etc. correctly.
; IT MUST BE INCLUDED
;
include "basicmac.r"

;
; let's start now
;
; Setting up the correct parameters for a routine.
; The header comments of a routine describe always the needed
; parameters and the restrictions for them. Thus, take a look
; at the routine files.
;
lea filepath(pc),a0
move.l a0,d0 ; file name
moveq #0,d1 ; no additional bytes to be allocated
moveq #0,d2 ; before and after the file

;
; the CALL_ macro makes the jump to the subroutine. It also tells
; the assembler that the subroutine is used and that it has to be
; assembler. Note that any unused subroutine of a routine file
; won't be assembler.
;
CALL_ LoadFile
move.l d0,fileaddr
move.l d2,fileblocklength

; ...

move.l 4.w,a6
move.l fileaddr(pc),a1
move.l fileblocklength(pc),d0
jsr _LVOFreeMem(a6)

rts

```

```
;
; At the end you have to include the routine files needed.
; They have to be included at the end; to be more precise, they must be
; included somewhere after the last access on one of their routines.
;
include "dosfile.r"

fileaddr: dc.l 0
fileblocklength:dc.l 0

filepath: dc.b "s:startup-sequence",0 ; just a file...
        even

; end-of-example
```

You may have also a glimpse at the example source codes that come along with the ProAsm Software. Some of them uses the routine files very heavily.

Possible Problems with the Evaluation Version of ProAsm

While assembling a source codes that needs any routine files, a 'REGISTERED VERSION ONLY: if you like ProAsm...' error may be issued. In the most cases this error was issued when an attempt was made to enable optimization, by commenting the concerned line (by inserting a semicolon (";") at its beginning) the optimization is suppressed and no error will be issued. Should this error be caused by any other reason, then you probably cannot assemble the source file with the evaluation version of ProAsm. Consider the 'registration.doc' for further details about limitations of the unregistered ProAsm version.

Author

If you have bugreports, questions, ideas, flames or complaints (constructive criticism is always welcome), or if you just want to contact me, write or send a letter to:

Daniel Weber

Internet: dweber@amiga.icu.net.ch (preferred)
 dweber@iic.ethz.ch

Mail: Daniel Weber
 Hoeflistrasse 32
 CH-8135 Langnau
 Switzerland.
