

++RCSRevision: 1.1 ++RCSDate: 1992/09/14 14:28:23

++catcode`++"=++active ++def"#1"++tt #1

Title: **A POSIX LIBRARY FOR THINK C**

Author: **TIMOTHY MURPHY**
TRINITY COLLEGE DUBLIN
("TIM@MATHS.TCD.IE")

++RCSDate
(Version ++RCSRevision)

begin abstract This library is intended to supplement the ANSI and Unix libraries provided with Think C, by supplying as many of the missing Posix functions as possible. The library was developed to assist in porting GNU programs to the Macintosh++footnotemark.

end abstract

++footnotetextTo date the author has ported the following GNU programs to the Mac: "diff-1.15", "patch-2.012u6", "rcs-5.6", "bison-1.18", "flex-2.3.7", "perl-4.035".

Status

This library is placed in the Public Domain. It may be used and modified freely---% though the author would be grateful for notification of any such modifications, and would prefer if his name and e-mail address were left on all files, to this end.

Availability

The library is available by anonymous FTP from "ftp.maths.tcd.ie" in the directory "pub/Mac/ThinkCPosix-1.1". The compiled library is contained in the file "Posix.hqx". The source files are archived in "ThinkCPosix.hqx". This includes the Think C project "Posix.++pi".

Acknowledgements

The author has used freely code in the archives pointed out to him, in particular code by: Guido van Rossum ("guido@cwi.nl"), and Mathias Neearcher. (Where code is taken more or less verbatim, this is indicated in the relevant files.)

He is also grateful for assistance from: Kenneth Seah ("kseah@procyon.austin.tx.us"), Sak Wathanasin ("sw@network-analysis-ltd.co.uk"), John W. Hardin ("hardin@mcc.com"), and Gary J. Henderson ("gary@iscnvx.lmsc.lockheed.com")

Posix

The use of the term ``Posix" may be misleading---% it is certainly not meant to imply that the functions satisfy any Posix standards (of which the author is largely ignorant). It is intended simply as a description of the kind of functions included---% namely, those Posix functions not included in the Think C ANSI or Unix libraries.

Many of the functions (in particular those in the file "dummy.c") do no more than return an appropriate value, indicating success or failure, as deemed appropriate.

Compilation

To re-compile the library, the included project "Posix.++pi" can be used. Just choose the "Build Library" option in the "Project" menu.

The author has all the ANSI options turned on, as well as "Require prototypes". The 4-byte int option and 68020 code option are turned off.

Usage

Any file using the library should
begin verbatim #include "ThinkCPosix.h"

end verbatim The header ("*.h") files should be placed somewhere in the Think C tree (ie below the Think C application).

Whenever the "Posix" library is included, the Think C "MacTraps" library must also be included. (This could probably be avoided by abstracting the very few MacTraps functions called. However, that is probably illegal?)

Notes These are a few random thoughts on some of the functions in the library.

"alloca()" This has been included to avoid duplication. Note that the prototype is "void *alloca(size_t)", rather than "char *alloca(unsigned)".

"dup()" The functions "dup()" and "dup2()" (as implemented here) both close the original file after duplication. The reason for this is that Think C adopts the non-standard, if plausible, policy of including some file-functions (such as "close") as part of the file structure. So a "close(fd)" addressed to the original descriptor would also close the duplicate. (This may be a misunderstanding on the author's part.)

"getpasswd()" This (and one or two other functions) call "getlogin()" in the Think C Unix library to find the user's name. This latter must be set---% under System 7, at least---% in the "Sharing Setup" Control Panel, as described in the Macintosh Networking Reference manual. (I believe that under System 6 it was changed on the Chooser panel.)

"Open()" The library includes a few ``substitute" functions, eg "Perror()" and "Open()". The idea is that these can be used in place of "perror()" or "open()" by including eg the line
begin verbatim #define open Open
end verbatim at a judicious point in the program (probably at the end of the configuration file, "conf.h" or whatever).

The function "Open(char*, int, ...)" allows "open()" to take more than 2 arguments (as seems increasingly the practice), although these extra arguments are ignored.

"Perror()" This gives a brief description of the error (if it can) rather than just the number as in "perror()". The Think C function "strerror()"---% which the author had not noticed---% may render this unnecessary.

Development

The author would be very grateful for any suggested improvements or extensions. Please send them to "tim@maths.tcd.ie". Any material used will be duly acknowledged.

Implementation of functions related to "pipe", "exec" and "spawn" would be particularly useful. (The "exec()" function in the Think C Unix library ignores all arguments except for the application to be launched. It would be nice---and must surely be feasible--- if these

arguments could be properly passed.)