

Troy's Kernel For Windows

Version 1.0

By Troy Rollo

Internet: troy@cbme.unsw.edu.au

Mail: Troy Rollo
Technical Director
Sycomp Pty Ltd
Level 4
1 James Place
North Sydney 2060
AUSTRALIA

µ

Troy's Kernel For Windows 1

Summary 3

Licensing 3

Limitations 3

Quick Start 3

TKERN Kernel Functions 4

ioctl 4

isatty 4

wait, wait3 and waitpid 4

kill 4

tkern_vallid_file 5

execv, execl 5

execv_ext, execl_ext 5

fork 5

tkern_total_zombies 5

tkern_list_zombies 5

tkern_get_process 5

tkern_is_device 5

Submitting Modifications 5

Summary

Troy's Kernel For Windows (aka TKERN) is a general purpose standard I/O library designed to make it easy to write and port command line oriented applications to Microsoft® Windows™. It is a replacement for both Borland's EasyWin, and Microsoft's QuickWin, and more.

Features include:

- fork emulation for fork/exec pairs
- wait, wait3 and waitpid calls
- kill call
- ability to have open files inherited by child processes
- parent/child relationship tracking
- built in command history mechanism in TKERN windows
- scrollbar in TKERN windows
- jump scrolling for faster output
- ioctl call to set parameters
- pipes (Real pipes, not the fake ones DOS has)

Licensing

Copyright © 1994 Troy Rollo <troy@cbme.unsw.EDU.AU>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details.

You should have received a copy of the GNU Library General Public License along with this library; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Limitations

TKERN was written for use with Borland C. Some modification to the stdio and links modules will be required to get it to work with Microsoft C or Symantec C.

No attempt has been made to port C++ format I/O.

Because the API interface of TKERN must remain consistent between versions, **TKERN.DLL** and **TKFMANGR.EXE** should never be compiled with a compiler other than Borland C. The definitions of any preprocessor constants are defined as the values given to those constants in Borland C++ 4.0.

Quick Start

To use this library:

- Copy **TKERN.DLL** and **TKFMANGR.EXE** to the **WINDOWS\SYSTEM** directory.
- Compile your program with the **TKERN INCLUDE** directory in the path before your compiler's standard **INCLUDE** directory
- Include the libraries **TKLIB.LIB** and **TKERN.LIB** in the link stage.

TKERN Kernel Functions

The names of functions in this section are the "external" names. These are the names you should use directly in your programs. Internally to **TKERN.DLL**, the implementation may have a name which includes the prefix "tkern_". The linkage, and any other adjustments required, are handled by trampoline functions in the **LINKS.OBJ** file in **TKLIB.LIB**.

The following functions behave exactly as documented in the BC++ library guide:

- open
- close
- read
- write
- lseek
- dup
- dup2

The remainder of this section lists added functions and functions with differing features.

ioctl

ioctl - I/O control.

The **TKERN ioctl** call behaves more like the UNIX version. The header files include most of the familiar UNIX tty functionality, although the only part of the UNIX functionality currently used is the window size subset of the terminal I/O control calls, including **TIOCGWINSZ**, **TIOCSWINSZ**, **TIOCGSIZE** and **TIOCSSIZE**.

An additional include file, `<sys/tkwin.h>` defines the following I/O control requests for **TKERN** windows.

- **WIOCGETHANDLE** returns the **HWND** associated with the file descriptor.
- **WIOCGETNAME** returns the title of the window.
- **WIOCSETNAME** sets the title of the window.
- **WIOCGETPARMS** returns the maximum number of lines in the scroll back buffer and history lists, as well as the number of lines to scroll for jump scrolling.
- **WIOCSETPARMS** sets the above.

See the file `<sys/tkwin.h>` for more details.

isatty

isatty - Return TRUE if the file descriptor is a TTY.

The TKERN version returns TRUE if the file descriptor is a TKERN window file descriptor.

wait, wait3 and waitpid

The behaviour of these is compatible with UNIX.

kill

The behaviour of kill is compatible with UNIX

tkern_valid_file

tkern_valid_file returns TRUE if the supplied file descriptor is valid and open.

execv, execl

The behaviour of these is compatible with UNIX

execv_ext, execl_ext

These are like execv and execl, except they take an additional parameter before the name of the program to execute. The additional parameter lists the command line arguments as a string, rather than as a vector. The primary use of this is so you can supply arguments and have them work with non TKERN applications.

fork

The tkern fork appears to behave as for the UNIX fork. The tkern fork must have a corresponding call to exec, and each exec call must have a corresponding call to fork. The one major difference is that an error in the exec call will be reflected as an error in the fork call, with the errno variable being set to the value of errno from the exec call.

fork/exec pairs may not be nested.

pipe

The pipe function takes as an argument a pointer to an array of two ints. The first gets filled with the read side of the pipe, and the second with the write side. TKERN will pause the write side process when the pipe is full, and will pause the read side process when the pipe is empty. There are no intermediate files, so you don't have to ensure sufficient disk space as you would if you were using a DOS pipe.

tkern_total_zombies

Returns the number of processes which have terminated whose parents have not yet waited for them.

tkern_list_zombies

Takes as arguments a pointer to an array of tk_process structures (defined in sys/task.h) and a number being the size of that array (in elements). Returns the number of entries written to the array.

tkern_get_process

Takes as arguments a pointer to a tk_process structure and an HTASK, and fills the tk_process structure with the information relating to that HTASK.

tkern_is_device

Takes a file descriptor and a pointer to a string as arguments. Returns TRUE if the file descriptor is open on the named device. Valid devices are currently **file** and **window**.

Submitting Modifications

Modifications may be submitted in one of the following forms:

- UNIX context diffs
- The entire source code after modification, provided the version on which the modifications are based is named.

Modifications should be submitted in machine readable form to either of the addresses at the start of this document.

While all modifications will be considered without prejudice, modifications will be rejected if:

- They change the formatting of existing code, no matter how ugly the existing code is. This tends to make things impossible to merge into the development copies.
- They contain too many changes to consider each change on its merits. It is better to separate patches into logical units with minimal numbers of features in each.
- The feature could be better implemented in another way. In this case you will be contacted and invited to implement it in the better way.

All submissions should include a valid email address on a network which is fully connected with the internet, or a telephone number.

TODO

Major obvious enhancements:

- Support for sockets (via Windows Sockets)
- A vt100 window device
- Support for communications ports as TTYs

To prevent wasted effort, please contact me before writing a major enhancement to ensure that nobody else is already writing the enhancement.

The TKERN mailing list

Pending sufficient interest, a mailing list will be created for TKERN developers. For subscriptions, send email to troy@cbme.unsw.EDU.AU.