The Non-Casual Reference to

PoSRIC

\ˈpoż-'rik

an acronym for the

Portable **S**cripted **R**iPorFS **I**nterface in **C**

By Charles Thompson

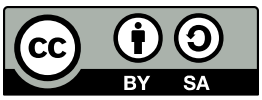


Table Of Contents

Introduction	.....	**I**
Building (Linux)	.....	**II.I**
Building (DOS)	.....	**II.II**
Commands	.....	**IIV**
RiPorFS vIIβ	.....	**IV.I**
RiPorFS Layout	.....	**IV.II**
RiPorFS Ridge #s	.....	**IV.IIV**

Introduction

RiPorFS was created with the idea of a truly limitless and portable file system. To make that idea a reality, **PoSRIC** has been created so you can access such filesystems on virtually any computer with a command line and a standard C library.

Building(Linux)

To build **PoSRIC** under linux you must have a GCC-like C compiler and some form of make. Once you have the compiler working, run `make` in the root **PoSRIC** directory to build the program, and the executable will be located in the local bin directory.

You can enable BASH colors by running `make cppargs=-DBASHCOLOR` instead of the above command

Building(DOS)

To build **PoSRIC** under DOS you must have Turbo C installed (Turbo C++ will probably work but I haven't tested it). Run the command `tc` in the root **PoSRIC** directory and press o+r+enter. Select the .TC that corresponds with your Turbo C version and press F9 to start the build process. Once building is done, press alt+x to return to the DOS prompt, where you can run `posric` to get started.

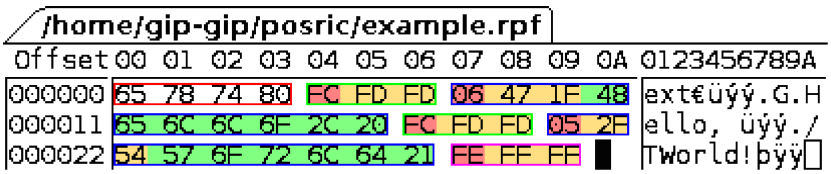
Commands

Command	Description
`#:foo is blue;`	Make a comment about foo being blue
`exit:;`	Exit posric with no errors
`giveUp:foo;`	Print “foo” and exit if the last command ended in error
`echo:foo;`	Print “foo” to the screen
`use:foo.rpf;`	Use the archive “foo.rpf”
`useName:foo`	Tells **PoSRIC** to apply future file-based commands to the file “foo”
`tmp:tmp.rpf;`	Use the file “tmp.rpf” as a temporary file
`format:;`	Format the archive being used
`list:;`	Lists all the files inside of the archive being used
`getFd:foo.bar;`	Gets filedaata from the in-archive file being used and stores it into “foo.bar”
`addFn:foo;`	Adds the filename “foo” to the archive, and gives an error if the file's detected
`addFd:foo.bar;`	Adds data from the file “foo.bar” to the in-archive file being used
`rmFile:foo;`	Removes the file “foo” from the archive, and gives an error if it doesn't exist

RiPorFS vIIß Specs

The **Ri**dged **Por**table **F**ile **S**ystem is a storage protocol based off of(as you guessed) ridges. Ridges are 8-bit unsigned integers paired with a fletcher16 checksum of the integer and(if it's a data ridge) the data that follows. The ridges have a dual purpose; to describe data, and to provide data. Description ridges have the 8th bit set, and describe the data ridges that follow. Data ridges encode blocks of data that are 1-128 bytes long, and are combined to be interpreted in relation to the preceding descriptor ridge .

RiPorFS Layout



Red border	Green border	Blue border	Pink border	Red fill	Tan fill	Green fill
signature	filename ridge	data ridge	end of fs ridge	ridge #	fletcher16 checksum	data

RiPorFS Ridge #s

Hex Value	Name	Description
FF	NULL Data	Used when the following data can be ignored
FE	End of FS Ridge	Used to signify the end of the filesystem
FD	File Data Ridge	Used when filedata is being read
FC	File Name Ridge	Used when a filename is being read. Encoded UTF-8
FB	Directory Name Ridge	Used when sorting files into a directory (named with the data being read). Encoded UTF-8
FA	Directory End Ridge	Used to end a directory
F9	Time Of Creation	Used to tell the time of creation of the following file/directory names. Data is a variable-length little-endian integer representing the time of creation in UNIX time.
F8	Time Of Modification	Used to tell the time of the last modification of the following file/directory names. Data is a variable-length little-endian integer representing the time of the last modification in UNIX time.
F7	File Owner	The name of the owner of the following files/directories. Encoded in UTF-8
F6	Last Writer	The name of the last person who modified the following files/directories. Encoded in UTF-8
F5	Permissions	The UNIX permissions for the following files/directories
F4	File Type	The MIME type of the following files

80	XML Metadata	Misc. XML metadata