Introduction to Windows File System Development

Tony Mason

Abstract

File Systems Development on Windows is different than on Linux or UNIX systems, both in the range of options and development environment. For all systems, developing a file system yields similar benefits by providing common services to a broad range of applications.

From a file systems development perspective, this tutorial provides a basic OS overview: Kernel, I/O Subsystem, Virtual Memory Manager, and Object Manager.

Windows naming differs from traditional POSIX operating systems so the tutorial discusses the name space, including 16-bit UNICODE character support and 32K maximum path names.

Additionally, the tutorial discusses the Windows kernel mode driver model, which allows dynamic loading and unloading of file system drivers. It discusses three distinct types of common file systems for Windows: physical media, network, and virtual (pseudo). It covers commonality and differences between the VFS interface and the Windows interface, for those interested in cross-platform support.

Further, the tutorial discusses *layered* file systems using “file system filter drivers”; while considered simpler to build, the pragmatic reality is that they are often more challenging to develop than traditional file systems.

In addition, the tutorial describes the Windows file system development environment including IDE, compilers, debuggers, installers, and conformance tests. It will describe the process of cryptographically signing file system drivers, which is a Windows requirement.

Finally, the tutorial will briefly describe FUSE on Windows, an alternative to constructing a kernel mode file system driver, which is highly compatible with the FUSE interface on UNIX and Linux systems.

Topics include:

- Windows kernel architecture
- I/O Subsystem
- Virtual Memory Subsystem

- Name space management
- File Systems Drivers
- File System Filter Drivers
- Development Tools
- Building, testing, validating, and deploying File System drivers
- FUSE on Windows as an alternative

Audience

This tutorial is designed for those interested in learning more about Windows kernel mode file systems development. Participants are expected to be familiar with operating systems fundamentals and have development experience. Upon completion, tutorial attendees will have a basic conceptual framework of how file systems fit into the Windows kernel. Tutorial attendees will be provided with source code that builds to a working kernel mode file system driver developed specifically for this session.

Author

Tony Mason is a PhD Student at the University of British Columbia, focusing on non-hierarchical file system name spaces and persistent memory. He has been developing file systems for the past 29 years, including on UNIX and Windows systems. Since 1996, he has taught numerous courses in Windows kernel mode development, including device drivers, kernel debugging, file systems, and file system filter driver development. Tony is co-author of *Windows NT Device Driver Development* [1] and *lex & yacc* [2].

Bibliography

[1] P. Viscarola and W. A. Mason, *Windows NT device driver development.* New Riders Publishing, 1998.

[2] J. R. Levine, J. R. Levine, T. Mason, and D. Brown, *Lex & yacc.* O'Reilly Media, Inc., 1992.