How Compliant is the Linux client?

Steve French
Azure Storage – Microsoft
Samba Team

SMB 3.1.1
Legal Statement

This work represents the views of the author(s) and does not necessarily reflect the views of Microsoft.

Linux is a registered trademark of Linus Torvalds.

Other company, product, and service names may be trademarks or service marks of others.
Outline

• Linux is a lot more than POSIX ...
• What works today?
  • Without POSIX Extensions
  • With POSIX extensions
• Some xfstesting details
• What to work on next?
• How to handle Linux continuing to extend APIs and test it ...
Linux > POSIX

- Currently huge number of syscalls!
  (try “git grep SYSCALL_DEFINE”
  well over 850 and 500+ are
  even documented “man syscalls”
  FS layer has 220). Verified today

vs

- Only about 100 POSIX API calls
Linux filesystems are not easy! Responsible for more than 200 of 850 syscalls. +4 since last year

<table>
<thead>
<tr>
<th>Syscall name</th>
<th>Kernel Version introduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>epoll_pwait2</td>
<td>5.11</td>
</tr>
<tr>
<td>mount_setattr</td>
<td>5.12</td>
</tr>
<tr>
<td>facessat2</td>
<td>5.8</td>
</tr>
<tr>
<td>close_range</td>
<td>5.9</td>
</tr>
</tbody>
</table>
goals: Fast! Easy! Transparent!

- Repeating an older slide about goals of SMB3.1.1:
  - Fastest, most secure general purpose way to access file data, whether cloud or on premises or virtualized
  - Implement all reasonable Linux/POSIX features - so apps don’t know they run on SMB3 mounts (vs. local)
  - As Linux evolves, and needs new features, quickly add to Linux kernel client and Samba and ksmbd
Why Not Other Protocols?

- SMB3.1.1 is easily extensible
- SMB3.1.1 works tightly with a set of protocols which can do more than any other file system protocol
- SMB3.1.1 has the best, most exhaustive set of testcases (not just smbtorture …)
- SMB3.1.1 and related protocols have more documentation (and documentation that has been tested and verified)
- SMB3.1.1 is proven across multiple client types, OS, architectures (and POSIX extensions have been a moving target, done before …)
- (And don’t forget … SAMBA rocks! And cifs.ko is one of most active FS)
What works today without POSIX Extensions

- Normal file and directory operations (open, read, write, fsync, close) to all servers, and hardlinks and even client handled symlinks (“mfsymlinks”), case preserving file name behavior, mapping almost all problematic characters in filenames (“" is the one exception)
- To most servers:
  - Sparse file operations: setsparse, query allocated ranges, punch hole
  - copy_range and clone_range (clone range is less commonly supported)
  - Special file handling via reparse points (or xattrs ala “sfu”)
  - Xattrs
- Emulation of mode bits via various alternatives (cifsacl, modefromsid)
What can be emulated today without POSIX Extensions

- Fcollapse and finsert
- Most delete and rename scenarios (some exceptions is where the rename fails with access denied with rename onto an existing file)
- Most byte range (easier with OFD rather than “posix” BRLs) and whole file lock scenarios
- Most of the special mode bits
What is problematic without POSIX Extensions

- Rename over an open file
- Files with pending delete showing up in the namespace (e.g. readdir)
- Case sensitive file names
- Locking scenarios that require advisory locks
- Populating a few fields in the statfs response (e.g. total and free inodes)
- Ownership can be preserved with “idsfromsid” on create, but only partially if using “multiuser,cifsacl” (especially important if accessing from multiple clients). When using cifsacl, user ownership is ok, but group ownership (gid) of a new file or directory will be the primary group specified by the user, which is not always the correct gid.
Quick Overview of POSIX Extensions Status

- Linux kernel client:
  - 5.1 kernel or later can be used but 5.8 or later recommended. Enable with mount option “posix.” All major features work on client.
    - readdir, create, mkdir, statfs, queryinfo (stat): complete
    - Support for new reparse tags for special files mostly complete (needs more testing)
- Samba (experimental tree available, enable with smb.conf parm)
  - Server
    - All major features work (thanks to JRA). Merge delayed due to time consuming conflicts with other large charges. Special file handling (Sockets, FIFOs, char device handling) needs to be updated
  - Client tools (smbclient)
    - Major features work. Additional options could be added to cmd set (Thanks to Volker)
- SMB3 Kernel server (cifsd's ksmbd.ko)
  - Partially implemented: it supports the POSIX negotiate context and partially parses POSIX open context
- 3rd party prototypes
- Wireshark patches available (network analysis)
POSIX Extensions Easy to Understand

- A simple negotiate context, an open context, a new file info level and a new fsinfo level
- Everything else relies on existing SMB3.1.1 features
<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Lenght</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>302</td>
<td>Negotiate Protocol Request</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>366</td>
<td>Negotiate Protocol Response</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>190</td>
<td>Session Setup Request, NTLMSSP_NEGOTIATE</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>400</td>
<td>Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_NEGOTIATE</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>480</td>
<td>Session Setup Request, NTLMSSP_AUTH, User: vsmfrench</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>142</td>
<td>Session Setup Response</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>176</td>
<td>Tree Connect Request Tree: \localhost\IPC$</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>176</td>
<td>Tree Connect Response</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>446</td>
<td>Create Request File: \GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>534</td>
<td>Create Response File: \GetInfo Response</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>191</td>
<td>_IOCTL Request FSCTL_QUERY_NETWORK_INTERFACE_INFO</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>143</td>
<td>_IOCTL Response, Error: STATUS_INVALID_DEVICE_REQUEST</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>182</td>
<td>GetInfo Request FS_INFO/FileFsAttributeInformation File:</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>150</td>
<td>GetInfo Response</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>175</td>
<td>GetInfo Request FS_INFO/FileFsDeviceInformation File:</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>180</td>
<td>GetInfo Response</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>175</td>
<td>GetInfo Request FS_INFO/FileFsVolumeInformation File:</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>158</td>
<td>Close Request File:</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>194</td>
<td>Close Response</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>224</td>
<td>_IOCTL Request FSCTL_DFS_GET_REFERRALS, File: \localhost\test</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>143</td>
<td>_IOCTL Response, Error: STATUS_NOT_FOUND</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>262</td>
<td>Create Request File:</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>354</td>
<td>Create Response File:</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>158</td>
<td>Close Request File:</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>194</td>
<td>Close Response</td>
</tr>
<tr>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>262</td>
<td>Create Response File:</td>
</tr>
<tr>
<td>NetBIOS Session Service</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMB2 (Server Message Block Protocol version 2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMB2 Header</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negotiate Protocol Request (0x00)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>StructureSize: 0x0024</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dialect count: 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security mode: 0x01, Signing enabled</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved: 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Capabilities: 0x00000077, DFS, LEASING, LARGE MTU, PERSISTENT HA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client Guid: 032f6fffc-4993-c44d-8b01-425c86949469</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NegotiateContextOffset: 0x0070</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NegotiateContextCount: 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved: 0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dialect: 0x0210</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dialect: 0x0300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dialect: 0x0302</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dialect: 0x0311</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negotiate Context: SMB2_PREAMUTH_INTEGRITY_CAPABILITIES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negotiate Context: SMB2_ENCRYPTION_CAPABILITIES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negotiate Context: Unknown Type: (0x5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Negotiate Context: SMB2_POSIX_EXTENSIONS_CAPABILITIES</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type: SMB2_POSIX_EXTENSIONS_CAPABILITIES (0x0100)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DataLength: 16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reserved: 00000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POSIX Reserved: 0x5025ad93</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NetBIOS Session Service

SMB2 (Server Message Block Protocol version 2)

SMB2 Header

Create Request (0x05)

- StructureSize: 0x0039
  - Oplock: No oplock (0x00)
  - Impersonation level: Impersonation (2)
  - Create Flags: 0x0000000000000000
  - Reserved: 0000000000000000
  - Access Mask: 0x00000100
  - File Attributes: 0x00000000
  - Share Access: 0x00000007, Read, Write, Delete
  - Disposition: Create (if file exists fail, else create it) (2)
  - Create Options: 0x00000001

Filename: 0760

- Blob Offset: 0x00000078
  - Blob Length: 8
- Blob Offset: 0x00000088
  - Blob Length: 40

ExtraInfo SMB2_POSIX_CREATE_CONTEXT

- Chain Element: SMB2_POSIX_CREATE_CONTEXT "5025ad93-b49c-e711-"
  - Chain Offset: 0x00000000
  - Tag: 5025ad93-b49c-e711-b423-83de960bcd7c
    - Blob Offset: 0x00000010
      - Blob Length: 16
    - Blob Offset: 0x00000020
      - Blob Length: 4
  - Data: POSIX Create Context request
    - POSIX perms: 0740
Some key problems with or without POSIX Extensions

- O_TMPFILE support
- Mapping of POSIX ACLs and RichACL on the wire
- SELinux integration
- Case sensitive xattr
- Better integration of Quota and Snapshot API with current Linux local fs tools (currently can be viewed with cifs-utils like “smbinfo”)
What Next?

- Examine the xfstest skips (and failures) in much detail and add small incremental changes
  
  • “xfstests” is the standard Linux fs functional test suite and no one file system can pass all tests due to various fs optional features.
  
  • Some can be emulated some need new flags

- Where that is not possible, consider adding new POSIX extensions version (simply adding additional uuid to the POSIX negotiate context)
What Next?

- What about minor extensions to reduce roundtrips and provide better/safer emulation?
  - Fcollapse and finsert are two examples
  - NTFS fsctls like FSCTL REARRANGE_FILE and SHUFFLE_FILE could help if available over SMB3
  - What about exposing Windows’s FILE_FLAG_POSIX_SEMANTICS
  - More compounding can help too
  - What about adding rename swap?
Examples from xfstest investigations

- Add support for renameat2 and rename exchange
- POSIX ACLs (can be emulated and there is pushback on implementing primitive POSIX ACLs)
- Support for additional chattr flags ("immutable" and "noatime" updates e.g.)
- fallocate –collapse-range
- Dedupe support
- Defragmentation support (may require VFS changes)
Examples from xfstest investigations

- Richacl support (tests 362 through 370) ??
- O_TMPFILE support (emulatable, but VFS changes would help)
- FITRIM support (may be emulatable)
- Quota support (may be emulatable already)
- Support for NFS export (nfs server on smb3 mounts)
- Case sensitive xattrs (EAs)
- SELinux support
Examples from xfstest investigations

- Support for online ‘label manipulation’ (see e.g. xfstest generic/492)
- Support for casefolding (“chattr +F”)
- Would native (rather than emulated) BSD flock (whole file lock) support help?
More details (with example xfstest #)

- atime options irrelevant (test 003)
- O_TMPFILE (generic/004)
- Defragmentation (018)
- Renameat2 (025)
- POSIX ACLs (026)
- FITRIM (038)
- Metadata journaling (049)
- Freezing fsctl (068) - https://lwn.net/Articles/287435/
- Chattr +ia (079) (“immutable”, “append only”)
- Chattr +A (277) (“no atime updates”)
- Linux disk quotas (082)
- Security (093) and trusted (097) xattr namespaces
- preallocated extent not marked with FIEMAP_EXTENT_UNWRITTEN (094)
- Dedupe (121)
- Advisory locks (131)
- suid/sgid bits are cleared after direct write (test 355)
- Richacls (362)
- Encryption support (395)
- Timestamp bounds unknown (402)
- chattr +d ("nodump") (424)
- Information about fiemap of attribute fork (425)
- NFS export (open by inode #) (426)
- Backslash in name ("Key urk???moo does not exist for FAKESLASH test??" in test 453)
More details (continued)

- Conflicting xattrs (test 454)
- XATTR_REPLACE (test 486)
- xfs_io label (492)
- Lsattr -d (508)
- Xattrs with slashes in name (523)
- Casefolding support (556)
- Dupremove utility (559), actton utility (596)
- Fsverity (571)
Next Steps

- Remember we can prototype to ksmbd as well now … and experiment …
Thank you for your time

- A very exciting time for ...

Linux + SMB 3