SMB3.1.1 POSIX Protocol Extensions: Summary and Current Implementation Status

Steve French
Azure Storage – Microsoft
Samba Team
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 ...
• Why do these extensions matter?
• Implementation Status
• What works today?
• Some details
• How to handle Linux continuing to extend APIs, to improve?
• Wireshark and Tracing
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 223). Verified today
  vs

- Only about 100 POSIX API calls
513 syscalls with man pages!
+12 just since last year’s SDC!

Some examples of new fs ones from past 9 months ...

<table>
<thead>
<tr>
<th>Syscall name</th>
<th>Kernel Version introduced</th>
</tr>
</thead>
<tbody>
<tr>
<td>io_uring_enter</td>
<td>5.1</td>
</tr>
<tr>
<td>io_uring_register</td>
<td>5.1</td>
</tr>
<tr>
<td>io_uring_setup</td>
<td>5.1</td>
</tr>
<tr>
<td>move_mount</td>
<td>5.2</td>
</tr>
<tr>
<td>open_tree</td>
<td>5.2</td>
</tr>
<tr>
<td>fsconfig</td>
<td>5.2</td>
</tr>
<tr>
<td>fsmount</td>
<td>5.2</td>
</tr>
<tr>
<td>fsopen</td>
<td>5.2</td>
</tr>
<tr>
<td>fspick</td>
<td>5.2</td>
</tr>
</tbody>
</table>
Repeating an old slide ...

Remember LINUX > POSIX
And not just new syscalls … new flags …

- 2 examples of richer Linux vs. simpler POSIX
- `fallocate` has 7 flags
  - Insert range
  - Unshare range
  - Zero range
  - Keep size
  - But POSIX `fallocate` has no flags
- Rename (renameat2) has 3 flags
  - noreplace, whiteout and exchange
  - POSIX rename has none
• Network File systems matter
  • these extensions to most popular network fs protocol (SMB3) are important
  • block devices struggle to do file system tasks: locking, security, leases, consistent metadata
• Linux Apps need to work over network mounts and continue to work as Linux evolves
• Improve common situations where customers have Linux and Windows and Mac clients
• Make sure extensions work with most secure, most optimal SMB3.1.1 dialect (don’t encourage less secure network file systems, or even SMB1/CIFS)
Quick Overview of Status

- **Linux kernel client:**
  - 5.1 kernel or later can be used. Enable with mount option “posix”
    - POSIX readdir support missing, and support for new reparse tags for special files (in progress)
- **Samba (experimental tree available, enable with smb.conf parm)**
  - Server
    - All major features work. 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
- **SMB3 Kernel server (cifsd’s ksmbd.ko)**
  - Early investigation
- **3rd party prototypes**
- **Wireshark patches available (network analysis)**
Why Samba?
Since 1992 …
Top Server on Linux

- Proven
- Broadly Implemented
- Extensible
- Secure
- Well Tested
- Implements richest File Protocol
- Enormous Client Base (Mac, Windows, now better on Linux! And more)
- 3.5 Million LOC!
<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>302</td>
<td>Negotiate Protocol Request</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>366</td>
<td>Negotiate Protocol Response</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>400</td>
<td>Session Setup Request, NTLMSSP_NEGOTIATE</td>
</tr>
<tr>
<td>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>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>142</td>
<td>Session Setup Response</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>170</td>
<td>Tree Connect Request Tree: \localhost\IPC$</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>170</td>
<td>Tree Connect Response Tree: \localhost\test</td>
</tr>
<tr>
<td>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>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>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>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>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>162</td>
<td>GetInfo Request FS_INFO/FileSystemAttributeInformation File:</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>150</td>
<td>GetInfo Response</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>175</td>
<td>GetInfo Request FS_INFO/FileSystemDeviceInformation File:</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>160</td>
<td>GetInfo Response</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>175</td>
<td>GetInfo Request FS_INFO/FileSystemVolumeInformation File:</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>158</td>
<td>GetInfo Response</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>194</td>
<td>Close Request File:</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>224</td>
<td>IOCTL Request FSCTL_DFS_GET_REFERALS, File: \localhost\test</td>
</tr>
<tr>
<td>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>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>262</td>
<td>Create Request File:</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>354</td>
<td>Create Response File:</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>158</td>
<td>Close Request File:</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>194</td>
<td>Close Response</td>
</tr>
<tr>
<td>0.0.1</td>
<td>127.0.0.1</td>
<td>SMB2</td>
<td>262</td>
<td>Create Request File:</td>
</tr>
</tbody>
</table>
NetBIOS Session Service

SMB2 (Server Message Block Protocol version 2)

- SMB2 Header
  - Negotiate Protocol Request (0x00)
    - StructureSize: 0x0024
      - Dialect count: 4
    - Security mode: 0x01, Signing enabled
      - Reserved: 0000
    - Capabilities: 0x00000077, DFS, LEASING, LARGE MTU, PERSISTENT HA
      - Client Guid: 032f6ffc-4993-c44d-8b01-425c86949469
      - NegotiateContextOffset: 0x0070
      - NegotiateContextCount: 4
        - Reserved: 0000
        - Dialect: 0x0210
        - Dialect: 0x0300
        - Dialect: 0x0302
        - Dialect: 0x0311
      - Negotiate Context: SMB2_PREAUTH_INTEGRITY_CAPACITYILITIES
      - Negotiate Context: SMB2_ENCRYPTION_CAPACITYILITIES
      - Negotiate Context: Unknown Type: (0x5)
    - Negotiate Context: SMB2_POSIX_EXTENSIONS_CAPACITYILITIES
      - Type: SMB2_POSIX_EXTENSIONS_CAPACITYILITIES (0x0100)
      - DataLength: 16
      - Reserved: 00000000
      - POSIX Reserved: 0x5025ad93
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-b3de060bcd7c
  - Blob Offset: 0x00000010
  - Blob Length: 16
  - Blob Offset: 0x00000020
  - Blob Length: 4
- Data: POSIX Create Context request
  - POSIX perms: 0740
Smbclient now has experimental support for SMB3.1.1 POSIX Extensions.

```
smfrench@smfrench-ThinkPad-P52:~$ /usr/local/samba/bin/smbclient //localhost/test -U testuser
Unable to initialize messaging context
Enter SAMBA\testuser's password:
Try "help" to get a list of possible commands.
smb: \> posix
SMB2 unix extensions supported
smb: \> posix_mkdir newdir 0777
posix_mkdir created directory \newdir
smb: \> help
?
  allinfo     alname     archive     backup
  blocksize   cancel     case_sensitive cd  chmod
  chown       close      del           deltree dir
  du          echo       exit          get    getfacl
  geteas      hardlink   help          history iosize
  lcd         link       lock          lowercase ls
  l           mask       md            mget    mkdir
  more        mput       newer         notify open
  posix       posix_encrypt posix_open posix_mkdir posix_rmdir
  posix_unlink posix_whoami print    prompt    put
  pwd         q           queue         quit     readlink
  rd          recurse     reget         rename    reput
  rm          rmdir       showacl       seteas   setmode
  scopy       stat        symlink       tar     tarmode
timeout     translate   unlock       volume   vuid
  wdel        logon       listconnect  showconnect tconn
ttdis        tid         utimes       logoff...
```

smb: \> posix_rmdir 0777
Failed to unlink directory \0777. NT_STATUS_OBJECT_NAME_NOT_FOUND
smb: \> posix_rmdir newdir
posix_rmdir deleted directory \newdir
smb: \> 
```
A year ago … and now …
kernel SMB3 client cifs.ko rapidly improving

- A year ago Linux 5.0-rc4 “Shy Crocodile”
- Last week: 5.5 “Kleptomaniac Octopus”
For cifs.ko (kernel client) to experiment with POSIX Extensions use 5.1 kernel or backport patch below

- commit 0d481325a9e5e3a31bf83bfcd3690a7a7152ece1
  Author: Steve French <stfrench@microsoft.com>
  Date: Sun Feb 24 17:56:33 2019 -0600
  smb3: Update POSIX negotiate context with POSIX ctxt GUID
An example from today (cifs.ko over SMB3.1.1 POSIX) mode bits preserved as expected on mkdir
Quality Much Improved – Top Priority

- More xfstests pass (> 150 and growing) even without POSIX extensions, vast majority of the rest are skipped due to missing features or being inappropriate for network file systems
- Many potential issues pointed out by static analysis addressed
- Starting 14 months ago The “Buildbot” … reducing regressions. VERY exciting addition for CIT (thanks Ronnie, Aurelien and Paulo)
- POSIX Extensions (jra’s tree) now a buildbot target for automated regression tests. Will be expanding test list run vs. it soon ...
POSIX: What could you try today?

• For obvious reasons these experimental changes not enabled by default:
  - With current mainline Linux (5.1 or later) must mount with "vers=3.1.1"
  AND also specify new mount option "posix" and turn off remapping of
  reserved characters (ie append "nomapposix")
  - Limited protocol features (posix open context request) can be tried but this
    small change VERY useful, enough to experiment with and test various apps

• JRA has a tree on samba.org (git.samba.org/jra/samba/.git in branch
  "master-smb2") with prototype server code

• Other vendors testing experimental distinct implementations of POSIX
  extensions as well
Why Isn’t This Shipped Already?

- Details are still being explored e.g recent interactions with WSL - Windows Subsystem for Linux
  - WSL defines methods of exporting filesystem types in reparse point tags we want to be compatible with on the wire.

- Samba VFS needs updating to match modern Linux/OpenGroup/POSIX APIs.
  - Samba VFS must change first to use Xxxat() (openat, mkdirat, linkat..) system calls before we can integrate this into Samba master.
What has changed?

- In order to optimize readdir better (to avoid extra roundtrips for each file):
  
  Symlinks and Special Files (char, block, fifo, socket) are now distinct reparse point tags (similar to what WSL does) rather than one tag for all (which requires an extra query for each special file)
New Proposed Tags for Special Files

IO_REPARSE_TAG_LX_SYMLINK
Symbolic link. Tag value is 0xA000001D.

IO_REPARSE_TAG_AF_UNIX
UNIX domain socket. Tag value is 0x80000023.

IO_REPARSE_TAG_LX_FIFO
FIFO. Tag value is 0x80000024.

IO_REPARSE_TAG_LX_CHR
Character special file. Tag value is 0x80000025.

IO_REPARSE_TAG_LX_BLK
Block special file. Tag value is 0x80000026.
Example: using it today

• On the client:
  • "mount –t smb3 //<address>/<share> /mnt -o
    username=<user>,password=<pass>,
    vers=3.1.1,posix,mfsymlinks,nomapposix,noperm"

• On the server add to smb.conf:
  • smb2 unix extensions = yes
  • "mangled names = no"
  • "directory mask = 07777"
  • "create mask = 07777"
  • Consider removing "obey pam restrictions"
Note that directory enumeration returns a reparse point tag (for file type).

- The proposed change to add tags this would reduce roundtrips
SMB3.1.1 POSIX Query Directory

```
<table>
<thead>
<tr>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0.0000...</td>
<td>127.0.0.1</td>
<td>127.0.0...</td>
<td>SMB2</td>
<td>470 Create Request File: filetypes;GetInfo Request FILE_INFO/SMB2_</td>
</tr>
<tr>
<td>3 0.0015...</td>
<td>127.0.0.1</td>
<td>127.0.0...</td>
<td>SMB2</td>
<td>678 Create Response File: filetypes;GetInfo Response;Close Response</td>
</tr>
<tr>
<td>5 0.0017...</td>
<td>127.0.0.1</td>
<td>127.0.0...</td>
<td>SMB2</td>
<td>278 Create Request File: filetypes</td>
</tr>
<tr>
<td>7 0.0031...</td>
<td>127.0.0.1</td>
<td>127.0.0...</td>
<td>SMB2</td>
<td>354 Create Response File: filetypes</td>
</tr>
<tr>
<td>9 0.0033...</td>
<td>127.0.0.1</td>
<td>127.0.0...</td>
<td>SMB2</td>
<td>168 Find Request File: filetypes (Level:0x64) Pattern: *</td>
</tr>
<tr>
<td>11 0.0058...</td>
<td>127.0.0.1</td>
<td>127.0.0...</td>
<td>SMB2</td>
<td>146 Find Response</td>
</tr>
<tr>
<td>13 0.0059...</td>
<td>127.0.0.1</td>
<td>127.0.0...</td>
<td>SMB2</td>
<td>168 Find Request File: filetypes (Level:0x64) Pattern: *</td>
</tr>
<tr>
<td>14 0.0061...</td>
<td>127.0.0.1</td>
<td>127.0.0...</td>
<td>SMB2</td>
<td>143 Find Response, Error: STATUS_NO_MORE_FILES</td>
</tr>
<tr>
<td>16 0.0063...</td>
<td>127.0.0.1</td>
<td>127.0.0...</td>
<td>SMB2</td>
<td>158 Close Request File: filetypes</td>
</tr>
</tbody>
</table>
```

Frame 11: 1406 bytes on wire (11248 bits), 1406 bytes captured (11248 bits) on interface 0
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
NetBIOS Session Service
SMB2 (Server Message Block Protocol version 2)
  SMB2 Header
    Find Response (0x6e)
      [Info Level: Unknown (100)]
      StructureSize: 0x0009
      Blob Offset: 0x00000048
      Blob Length: 1264
      Info: 80000000000000056f4d36f5072d501a6cb5705072d501...
      Unknown: 800000000000000056f4d36f5072d501a6cb5705072d501...

Summary of What works

• Without Extensions

• With Extensions
Example (w/o POSIX extensions)
(thank you to Aurelien at SuSE!)

- Mode bits (in special ACE)
  - Mode bits on left
  - File name on right
- NB mkfifo not finished
- Ownership works too

444 mnt/smb/0444
540 mnt/smb/0540
777 mnt/smb/0777
1777 mnt/smb/1777
2777 mnt/smb/2777
3777 mnt/smb/3777
1777 mnt/smb/d01777
2777 mnt/smb/d02777
444 mnt/smb/d0444
4777 mnt/smb/d04777
540 mnt/smb/d0540
777 mnt/smb/d0777
644 mnt/smb/emptyfile
stat: cannot stat 'mnt/smb/fifo': Operation not supported
644 mnt/smb/file-as-sfrench
• What about the Linux Kernel?
  
  - New API changes added about once a year to the VFS (minor global changes added more often, but not all could affect what we need to send on the wire in perfect world ...)
  
  - Need to quickly update protocol when not possible to do over SMB3
  
  - Need better interaction with key communities (containers, databases and many others) about what they would like to see
SMB3 POSIX Extensions

- Negotiate Protocol
  - SMB3.1.1 (or later required)
    - POSIX Negotiate Context 0x100
      - Version is included in the context by including the GUID of the supported POSIX open context(s) – currently only one supported
  - If POSIX open contexts not supported, negotiate context must be ignored
  - If POSIX open contexts supported for some files then negotiate context is returned, but server must fail opens with POSIX contexts for files where POSIX is not supported (rather than ignoring the POSIX context)

- Tree Connect – in future dialects tree connect contexts may allow more granularity in allowing servers to tell clients which shares they can't use POSIX opens on

- Case sensitivity yes/no can be exposed via existing QFS Info call
POSIX Extension
Requirements

• If server returns a POSIX create context on an open:
  - It supports case sensitive names on this path
  - It supports POSIX unlink/rename semantics on this file
  - It supports advisory (POSIX) locking on this file.
    • Actually they are “OFD” not “POSIX” locks (see e.g.
      https://gavv.github.io/blog/file-locks/#emulating-open-file-descriptio
      n-locks)
  - PATH names are not remapped (no SFU remap needed for
    * and \ and > and < and : …). UCS2 converted directly to
    UTF-8 and server supports POSIX pathnames
We Leverage Existing SMB3 features

- Hardlinks use Windows setinfo call (long ago implemented)
- Symlinks, mkfifo, mknod use “nfs-reparse point” (MS-FSCC 2.1.2.6) – Distinct reparse point tags for each type of special file (ala what “WSL” does) - allows us to better optimize readdir
- ACE with special SID (with mode at end) ala “NFS ACL” mapping can be used to set mode (SID: S-1-5-88-3) see http://people.redhat.com/steved/Bakeathon-2010/SDC2010-NFS-Windows.jbiseda.20100921.pdf
- Other linux extensions, e.g. fallocate are mapped to existing SMB3 operations where possible
SMB311 POSIX Extensions

• Create/Open
  - New POSIX create context
    • If POSIX supported then context must be returned on all opens for which POSIX create context was sent (or open should be failed)
    • It is allowed to have POSIX and non-POSIX opens on the same file
    • It is allowed to have some files in a server which are POSIX and some which are not
Format of the POSIX owner and mid information in the ACL

<NTSecurityAuthority>-<SECURITY_NFS_ID_BASE_RID>-<NfsSidType>-<NfsSidValue>

Owner SID for UID: “S-1-5-88-1-<uid>”
Group SID for GID: “S-1-5-88-2-<gid>”
Mode SID: “S-1-5-88-3-<mode>”
Everyone: “S-1-5-88-4”
POSIX Infolevels

- Query/SetInfo and Query_DIR (and also FSInfo)
  - Level 0x64  SMB2_FIND_POSIX_INFORMATION
  - Payload variable (Max = 216 bytes)
    - Timestamps
    - File size
    - Dos attributes
    - U64 Inode number
    - U32 device id
    - U32 zero
    - Struct posix_create_context_response
Wireshark

• See Aurelien’s dissector improvements
  - https://github.com/aaptel/wireshark/commits/smb3unix
  - And Pike sample test code
    • https://github.com/aaptel/pike/tree/smb3unix
Next Steps

• Continue debugging test implementations (cifs.ko and JRAs Samba POSIX test branch). Current focus (eg at this test event): enhancing smb3 client to better handle POSIX readdir

• Continue to add xfstests to the ‘jraposix’ test group in the buildbot (to regression test the client against Samba server with POSIX extensions)

• Examine EVERY xfstest skip and every xfstest fail for potential match to features in (or to add to) SMB3.1.1 POSIX Extensions
  • e.g. enabling special files (fifos, blkdevs etc.) as reparse points enables five tests.

• Continue extending the wireshark dissectors (see Aurelien)

• Continue testing here

• Continue updating the wiki with details:
  
  https://wiki.samba.org/index.php/SMB3-Linux

• Questions/comments welcome: samba-technical and linux-cifs lists
Thank you for your time

- This is a very exciting time for ...

Linux + SMB 3
POSIX Path Names Work

root@smf-Thinkpad-P51:~# ls /posix-extensions-mount/
  d0754  fileacolon:  'fileexclamation!'  hello
  'file!' 'fileasterisk*' 'filequestion?'  newfile
root@smf-Thinkpad-P51:~# ls /scratch
  d0754  fileacolon:  'fileexclamation!'  hello
  'file!' 'fileasterisk*' 'filequestion?'  newfile
root@smf-Thinkpad-P51:~# uname -a
Linux smf-Thinkpad-P51 5.0.0-rc4+ #67 SMP Sun Jan 27 20:49:32 019 x86_64 x86_64 x86_64 GNU/Linux
root@smf-Thinkpad-P51:~# ~/posix/bin/smbd -V
Version 4.10.0pre1-DEVELOPERBUILD
Note the new mount option “posix” (vs “nounix”)
Mode bits on create and case sensitivity work!
Rename works with POSIX extensions!
Statfs ("stat –f") without POSIX extensions:

```
root@smf-Thinkpad-P51:~:/cifs-2.6# cat /proc/mounts | grep cifs
//localhost/scratch /mnt cifs rw,relatime,vers=3.0,cache=strict,username=testuser,domain=,uid=0,noforceuid,gid=0,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,nouser,x,serverino,mfsymlinks,noperm,rsize=1048576,wsize=1048576,echo_interval=60,actimeo=1 0
root@smf-Thinkpad-P51:~:/cifs-2.6# stat -f /mnt
  File: "/mnt"
    ID: 0        Namelen: 4096   Type: smb2
  Block size: 1024         Fundamental block size: 1024
  Blocks: Total: 234804176  Free: 28323720          Available: 28323720
  Inodes: Total: 0         Free: 0
root@smf-Thinkpad-P51:~:/cifs-2.6# stat -f /scratch
  File: "/scratch"
    ID: e94471edc7140504 Namelen: 255   Type: ext2/ext3
  Block size: 4096         Fundamental block size: 4096
  Blocks: Total: 58701044  Free: 10080212          Available: 7080929
  Inodes: Total: 14983168  Free: 13901548
```
Statfs ("stat –f") with POSIX extensions – works!

```
root@smf-Thinkpad-P51:~/cifs-2.6# cat /proc-mounts | grep smb
//127.0.0.1/scratch /mnt1 smb3 rw,relatime,vers=3.1.1,cache=strict,username=testuser,devuid=0,noforceuid,gid=0,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,prot=6,squash,allow_root,allow_other,prevent_suid,uid=0,umask=0,sync,softlinks,hardlinks,forwserve,allow_otheruid,allow_othergid,notrace,sec=systemd,nosuid,soft seventy_percent,softlimit
root@smf-Thinkpad-P51:~/cifs-2.6# stat -f /mnt1
  File: "/mnt1"
    ID: 0      Namelen: 4096      Type: smb2
Block size: 4096      Fundamental block size: 4096
Blocks: Total: 58701044  Free: 10080249  Available: 7080966
Inodes: Total: 14983168  Free: 13901538
root@smf-Thinkpad-P51:~/cifs-2.6# stat -f /scratch
  File: "/scratch"
    ID: e94471edc7140504  Namelen: 255      Type: ext2/ext3
Block size: 4096      Fundamental block size: 4096
Blocks: Total: 58701044  Free: 10080127  Available: 7080844
Inodes: Total: 14983168  Free: 13901538
```
Details – Negotiate Req
(w/POSIX)

- Format has changed since 2018 SDC
- Now 16 byte GUID
- Allows versioning
Details continued – create response