Accessing files remotely from the smallest devices to the largest devices (and the cloud): SMB3.1.1 Improvements to the Linux client
- This work represents the views of the author(s) and does not necessarily reflect the views of Microsoft Corporation
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.
Who am I?

- Steve French smfrench@gmail.com
- Author and maintainer of Linux cifs vfs (for accessing Samba, Azure, Windows and various SMB3/CIFS based NAS appliances)
- Co-maintainer of the new kernel server (ksmbd)
- Also wrote initial SMB2 kernel client prototype
- Member of the Samba team
- Coauthor of SNIA CIFS Technical Reference, former SNIA CIFS Working Group chair
- Principal Software Engineer, Azure Storage: Microsoft
Outline

- Overview of Linux FS activity
- Recent ksmbd (server) improvements
- Recent client improvements
- Coming soon … what to look forward to
- Testing improvements
A year ago and now ...

- Now: 6.3 “Hurr durr I’m a ninja sloth”
- Then: 5.18-rc4 “Superb Owl”
LSF/MM/eBPF summit is back in person too

- Picture from 2022 (2023 summit is going on same week as SambaXP)
Some Linux FS topics of interest from LSF and other recent discussions

- Folios, netfs, iov_iter, variable size pages, and the redesign of page cache and offline (fscache)
- Improvements to statx and fsinfo and to inotify/fanotify
- Idmapped mounts
- Updates to POSIX ACL internal API
- Extending in kernel encryption: TLS handshake (for NFS) and QUIC (SMB3.1.1 and other)
- io_uring (async i/o improvements)
- Shift to cloud
- Better support for faster storage (NVME) and net (RDMA/smbdirect)
Linux Filesystems Activity over past year (since 5.18-rc4)

- 5400 filesystems changesets (6.2% of total kernel changesets, one of the most watched parts of the kernel, and FS activity is up slightly)
- Linux kernel fs are 1.07 million lines of code (measured this week)
Most Active Linux Filesystems over the past year

- VFS (mapping layer) 420 changesets
- The top filesystems and VFS dominate the activity
- Most active are BTRFS 1216 (huge increase), XFS 553, ext4 386
- SMB3.1.1 (cifs.ko) 339 (activity up)
- Then NFSD (server) 279, and NFS (client) 208 (activity down)
  - cifs.ko had more than 3x more lines changed. It has been a VERY active year for cifs.ko
- Other:
  - ksmbd (new, added in the 5.15 kernel) (121), ntfs3 (178, added in 5.15), gfs2 (129), ceph (99)
SMB3.1.1 Activity was strong this year

- cifs.ko activity was strong, 339 changesets
  - cifs is 60KLOC kernel code (not counting user space utilities)

- ksmbd activity was down
  - Introduced in the 5.15 kernel, 25KLOC kernel code, 310 changesets since its introduction

- Samba server (userspace) is over 3.5 million lines of code (orders of magnitude bigger than the kernel smbd server or any of the NFS servers) and is even more active
Goals and Actions for SMB3.1.1 on Linux

- Be the fastest, most secure general-purpose way to access file data, whether in the cloud or on premises or virtualized
  - Improve directory lease support
  - Keep improving compounding, multichannel
- Support more Linux/POSIX features – so apps don’t know they run on SMB3 mounts (vs. local)
  - SMB3.1.1 POSIX extensions, new FSCTLs
  - Use xfstests to locate new features to emulate
- As Linux evolves, quickly add features to Linux kernel client and Samba and ksmbd
  - More test automation and keep adding more tests
One of the strengths of SMB3.1.1 is broad interop testing

- In-person plugfests are back!
- SMB3.1.1 plugfest collocated with SDC last fall
- Hoping for much informal testing here
  - Contact Paulo and Enzo e.g.
- Many exciting things being tested
Progress and Status update for Linux Kernel Server (ksmbd)

Additional information provided by Namjae Jeon (linkinjeon@kernel.org)
Architecture

User Space

ksmbd.mountd

Share configuration
ID/PW configuration

DCE/RPC

Netlink Interface

ksmbd/0
(forker thread)

Kernel Space

Socket (445)

ksmbd/1
ksmbd/2
ksmbd/N

ksmbd.adduser

ksmbdpwd.db (ID/PW file)

smb.conf (config file)

Authentication

NTLM
NTLMv2

kerberos

Dialects

SMB2.1
SMB3
SMB3.1.1

Client

Local Filesystem

KERNEL
Some examples of exciting recent progress

• POSIX extensions
  • Server supports SMB3.1.1 POSIX Extensions
  • Change the SID to the one Samba server using for POSIX Extensions
    • Samba set SIDOWNER and SIDUNIX_GROUP in create posix context
    • And sets SIDUNIX
  • Set file permission to match Samba server POSIX extension behavior
  • Fill in SIDs in SMB_FIND_FILE_POSIX_INFO responses
• Fixes for various security issues
  • ZDI and others had reported several security issues
• Fix unlink and rename races (new Linux VFS helpers added to aid this)
• Multichannel and SMB Direct improvements
• Improve management of SMB3 credits (flow control improvements)
RSS (Receive Side Scaling) mode support

- ksmbd now supports RSS mode
- Ziwei Xie (high-flyer) compared the performance of Samba and ksmbd on their test environment. Thanks Ziwei!
- In RSS mode, there is a performance difference of 3 times for read and 4 times for write on his setup.
Performance Comparison on Multichannel + RSS mode

*Test Environment
CPU : AMD EPYC 7H12 64-Core Processor
NIC : MCX653105A-HDAT
Client : Windows
Benchmark tool : FIO

![Chart showing performance comparison between read and write operations.](chart.png)

- Read performance improved by +290%
- Write performance improved by +550%
Future Plan

• More Directory lease testing
  • Plan to turn on leases by default (instead of older oplocks, current default)
• SMB2 notify (WIP)
• Improve MacOS compatibility
• Add new FSCTLs to help Linux kernel client
  • e.g. rename exchange and any remaining fallocate corner cases
• Durable handle v1/v2 feature (WIP)
• Add ksmbd status option to show statistics using ksmbd.control
  • Processed requests, session info(user info, number of credits and more), session list, openfiles, NIC info.
• Config backend (Recently get a request, make the configuration interface available remotely over the WINREG RPC interface)
Linux Kernel Server, KSMBD (continued)

- If interested in contributing there are lots of cool features to work on, as well as improved integration with Samba (e.g. user space upcalls for additional features). The SMB3.1.1 family of protocols is huge!
- Roles: Namjae (the maintainer) has done a lot, but additional features or subcomponents could be delegated. I am managing the git merges, ensuring additional functional testing is done regularly, and reviewing patches as requested by Namjae (my focus is largely on the client)
- Namjae would welcome additional help with code reviews, security auditing, testing and new features
- Very exciting time!
Recent improvements in the kernel client

(cifs.ko)
Signing algorithm negotiation – faster signing

```
# modinfo cifs | grep signing
parm: enable_negotiate_signing:Enable negotiating
packet signing algorithm with server. Default: n/N/0 (bool)

"insmod cifs.ko enable_negotiate_signing"
```
6.0 and earlier kernels /proc/fs/cifs/DebugData showed:
   Security type: RawNTLMSSP  SessionId: 0x5f08b08 signed

About 20% faster performance was demonstrated if workload not network constrained (thank you Enzo!). Still testing AES-GMAC

   Security type: RawNTLMSSP  SessionId: 0x5f08b08 signed (AES-GMAC)

Or if server doesn’t support GMAC will fall back to:
   Security type: RawNTLMSSP  SessionId: 0x5f08b08 signed (AES-CMAC)
Example perf #s

- “dd if=/dev/zero of=/mnt/target bs=4M count=256”
- Signing (default prior to patches): 280MB/sec
- Signing (GMAC, with the experimental patches): 310MB/sec
- Encryption (vers=3.0, CCM): 170MB/sec
- Encryption (vers=3.1.1 GCM): 1.1GB/sec
- (testing on my laptop at SDC)
Directory Caching Improvements

• Thanks to Ronnie Sahlberg, directory caching and use of directory leases (to improve metadata caching even more, and safely) is MUCH improved

• Huge perf win!

• Continuing to optimize
Notice cached directory information with lease reduces requests needed (stat does not need to be sent)
<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.049776613</td>
<td>172.38.33.95</td>
<td>172.38.32.1</td>
<td>SMB2</td>
<td>416</td>
<td>Create Request File: tmp;GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO;Close Request</td>
</tr>
<tr>
<td>3</td>
<td>0.049822339</td>
<td>172.38.32.1</td>
<td>172.38.33.95</td>
<td>SMB2</td>
<td>584</td>
<td>Create Response File: tmp;GetInfo Response;Close Response</td>
</tr>
<tr>
<td>5</td>
<td>0.04879215</td>
<td>172.38.33.95</td>
<td>172.38.32.1</td>
<td>SMB2</td>
<td>408</td>
<td>Create Request File: tmp;GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO</td>
</tr>
<tr>
<td>6</td>
<td>0.049323137</td>
<td>172.38.32.1</td>
<td>172.38.33.95</td>
<td>SMB2</td>
<td>556</td>
<td>Create Response File: tmp;GetInfo Response</td>
</tr>
<tr>
<td>7</td>
<td>0.049406653</td>
<td>172.38.33.95</td>
<td>172.38.32.1</td>
<td>SMB2</td>
<td>322</td>
<td>Create Request File: tmp;Find Request SMB2 FIND ID FULL DIRECTORY_INFO Pattern: *</td>
</tr>
<tr>
<td>9</td>
<td>0.050158248</td>
<td>172.38.33.95</td>
<td>172.38.32.1</td>
<td>SMB2</td>
<td>179</td>
<td>Find Request File: tmp SMB2 FIND ID FULL DIRECTORY_INFO Pattern: *</td>
</tr>
<tr>
<td>10</td>
<td>0.050498847</td>
<td>172.38.32.1</td>
<td>172.38.33.95</td>
<td>SMB2</td>
<td>144</td>
<td>Find Response, Error: STATUS_NO_MORE_FILES</td>
</tr>
<tr>
<td>11</td>
<td>0.050653669</td>
<td>172.38.33.95</td>
<td>172.38.32.1</td>
<td>SMB2</td>
<td>160</td>
<td>Close Request File: tmp</td>
</tr>
<tr>
<td>12</td>
<td>0.050948621</td>
<td>172.38.32.1</td>
<td>172.38.33.95</td>
<td>SMB2</td>
<td>190</td>
<td>Close Response</td>
</tr>
<tr>
<td>14</td>
<td>0.057595339</td>
<td>172.38.33.95</td>
<td>172.38.32.1</td>
<td>SMB2</td>
<td>160</td>
<td>Close Request File: tmp</td>
</tr>
<tr>
<td>15</td>
<td>0.057054189</td>
<td>172.38.32.1</td>
<td>172.38.33.95</td>
<td>SMB2</td>
<td>190</td>
<td>Close Response</td>
</tr>
</tbody>
</table>

**SMB2 (Server Message Block Protocol version 2)**

- **SMB2 Header**
  - Create Response (0x65)
    - StructureSize: 0x0005
    - Oplock: No oplock (0x00)
    - Response Flags: 0x06
      - Create Action: The file existed and was opened (1)
      - Create: Feb 21, 2022 14:21:08.407828580 CST
      - Last Access: Sep 14, 2022 11:38:54.584345900 CDT
      - Last Write: Feb 21, 2022 14:21:08.407828580 CST
      - Last Change: Feb 21, 2022 14:21:08.407828580 CST
      - Allocation Size: 0
      - End Of File: 0
      - File Attributes: 0x00000018
        - Reserved: 00000000
      - GUID handle File: tmp
        - Blob Offset: 0x00000000
        - Blob Length: 56
      - ExtraInfo SMB2_CREATE_QUERY_ON_DISK_ID

- **SMB2 (Server Message Block Protocol version 2)**
  - SMB2 Header
    - Find Response (0x60)
      - [Info Level: SMB2_FIND_ID_FULL_DIRECTORY_INFO (38)]
        - StructureSize: 0x0069
        - Blob Offset: 0x00000040
        - Blob Length: 232
      - Info: 5800000000000000512728c602780112ca202b588d801e685728c6027d801e685728c...
        - FileIdBothDirectoryInfo: ...
        - FileIdBothDirectoryInfo: ...
        - FileIdBothDirectoryInfo: populate_root
Deferred close improvements

- Currently used for i/o patterns like open/read/close/open/read/close
- Extending to cover many more scenarios, greatly improving performance
- Handle cache (deferred close time) now configurable with new mount parm “closetimeo” (thank you Bharath!)
- Improvements to lease break corner cases recently added
SMB1/CIFS deprecation

• SMB3.1.1 rocks …
• Gradually move the old, insecure dialects out of the default module used for SMB2.1/SMB3/SMB3.1.1, so easier to deprecate SMB1/CIFS
Multichannel improvements

• Requerying network interfaces, dynamically adjusting
• Reconnect improvements
• Performance improvements (thank you Shyam Prasad!)
• Soon will be enabled by default (when server supports multiple interfaces or RSS)
SMB Direct (thanks to Tom Talpey and Metze)

- Reduce SGE usage, and decrease maximum fragment size
  - Fails to operate on SoftiWARP provider
  - Needless memory usage
  - High SGE usage impacts performance
- Fix RDMA “responder resources”, which do NOT apply to RDMA Writes
  - Significant performance limiter for bulk reads
- Fix sends to **not wait for completion** before returning
  - Stalls the pipeline, and costs significant context switching
- Use RDMA post-multiple to improve compound send efficiency
- Ensure packet kmem cache optimal packing (3x1364 == 4092)
- Review protocol parsing and state validation
  - E.g. ksmbd allows renegotiate (?), reassembles oversize segments (?)
- Hangs when shutting down with connection held
- Merge the two implementations: fs/cifs/smbdirect.[ch] and fs/ksmbd/transport_rdma.[ch]
  - Either refactor and merge, or consider metze’s alternative “smbdirect socket” driver
SMBDIRECT Transport Improvements – RDMA for the world

• SMBDIRECT is an abstraction layer for making RDMA useable more broadly. It has no SMB dependencies (SMB3 was just the first consumer of this generic transport layer, but it applies more broadly)

• Longer term plan is to:
  • Bring common from cifs.ko and ksmbd for RDMA into smbdirect.ko
  • Enable user space access to RDMA through smbdirect.ko so user space applications can benefit from the performance gains of RDMA
  • Improvements to this common module will benefit both client and server (and userspace)

• smbdirect.ko will provide
  • PF_SMBDIRECT sockets
  • Send message and receive message will get MSG_OOB messages for read and write offload, greatly improving performance and reducing CPU overhead

• (SMB independent) “echo server client” smbdirect tests under development to improve regression testing without requiring SMB

• Thanks to Metze for this work. Feedback and review and testing welcome.
SMB3.1.1 POSIX Extensions

See my other talk at SambaXP and Volker’s talks
Has been in Linux client for years & is simpler than SMB1 Unix extensions
Great progress on Samba (client and server) and ksmbd server
Testing now possible with three servers and at least two clients
Setting up Samba and ksmbd shares are easy. Samba requires “smb3 unix extensions = yes” in smb.conf.

```
[global]
  workgroup = SAMBA
  map to guest = Bad User
  passdb backend = tdb
  printing = cups
  printcap name = cups
  host msdfs = yes
  server multi channel support = yes
  log level = 4
  smb3 unix extensions = yes

[scratch]
  comment = scratch share for testing
  browseable = yes
  path = /scratch
  guest ok = yes
  read only = no
  ea support = yes
  create mask = 0777
```

```
Note exact mode bits and owner reported w/POSIX Extensions
Query FS Info
– includes additional posix fields

```
root@smfrench-ThinkPad-P52:/home/smfrench# mount -t cifs //local -o username=testuser, password=testpass, mfsymlinks, posix
root@smfrench-ThinkPad-P52:/home/smfrench# stat -f /mnt1
File: "/mnt1"
  ID: c7df5aa0f1e89eff Namelen: 255   Type: smb2
  Block size: 4096   Fundamental block size: 4096
Blocks: Total: 139092115  Free: 48993190   Available: 
Inodes: Total: 278320128  Free: 273211966
root@smfrench-ThinkPad-P52:/home/smfrench#
```
Now better performance (POSIX QFS Info now compounded)
5.17 kernel (March 20th), 51 changesets, cifs.ko ver 2.35

- Add support for new fscache (offline files caching mechanism)
- Send additional NTLMSSP info (including module and OS version) for improved debugging
- DFS and ACL fixes
- Key modefromsid fix (where client enforced mode bits retrieved from special ACE)
- Restructuring of multichannel code
5.18 kernel (May 22\textsuperscript{nd}), 40 changesets, cifs.ko ver 2.36

- Important performance improvement (reuse cached file handle for various common operations like stat and statfs if available), greatly reducing metadata operations (like open/close)
- Important fscache (offline file caching) and DFS improvements
- cross mount refile link now supported, which can dramatically improve copy performance from one share to another (on the same server) if they support duplicate extents.
5.19 kernel (July 31\textsuperscript{st}, 2022)

- Important performance optimization for directory searches, now we cache the root directory content (to the many servers which support directory leases) reducing amount of network traffic for queries in the root directory
- Multichannel reconnect improvements (e.g. when address or interfaces change)
- RDMA (smbdirect) improvements
- New mount parm “nosparse” to optionally disable use of sparse...
Fallocate improvements (insert and collapse range)
Module size shrunk significantly when SMB1/CIFS (insecure legacy) disabled
New mount parm “closetimeo” allows extending deferred closes (handle leases) longer or even disabling the feature (and default increased to 5 seconds from 1 sec)
Important deferred close fix
Multichannel perf (locking) improvements
6.1 kernel (Dec 11th, 2022) (cifs module ver: 2.40)

- Performance improvement for path revalidation (metadata ops perf better) by using cached dentry for subdirectory if lease held on it
  - Expanding cached directories to include subdirectories (thanks Ronnie!)
- New ioctl for change notify added that returns the name(s) of any changed files in the directory (not just that the directory has changed)
  - e.g. so app can do their own offline caching of files and sync with server
- Improve symlink handling (avoid an extra roundtrip when symlink detected via STOPPED_ON_SYMLINK message)
- RDMA (smbdirect) improvements (thanks Tom Talpey and M...
6.2 kernel (February 19\textsuperscript{th}, 2023) (cifs module 2.41)

- Important SMB3.1.1 POSIX extensions improvement (parse owner and group SIDs to improve stat output)
- DFS performance improvements (reducing roundtrips) and DFS fixes
- Multichannel improvements
- Integration with the new kernel page caching infrastructure, folios, iov_iter and memory management layering cleanup
6.3 kernel (April 23rd, 2023) (cifs module 2.42)

- Kernel idmapping improvements
- Improvements to use folios (better mm integration and cached writes)
- RDMA (smbdirect) improvements (thanks Metze and David)
- Many multichannel improvements (including using least loaded channel for sending I/O, and improvements for reconnect). Thanks Shyam!
- Various DFS fixes
- Lower default deferred close timeout
6.4-rc kernel (expected early July) (cifs.ko version: 2.43)

- Important deferred close (lease break corner case) fixes
- Reconnect and DFS fixes
- Important crediting improvements expected
- Compounding improvements expected
Tracing continues to improve ...

- Added 4 additional dynamic tracepoints

```
root@smfrench-ThinkPad-P52:/sys/kernel/tracing/events/cifs# ls
smb3_flush_err  smb3_fsync_err  smb3_posix_mkdir_enter
enable          smb3_hardlink_done
smb3_add_credits smb3_hardlink_err
smb3_adj_credits smb3_insufficient_credits
smb3_close_done  smb3_lease_done
smb3_close_err   smb3_lease_err
smb3_cmd_done    smb3_lease_not_found
smb3_cmd_enter   smb3_lock_err
smb3_connect_done smb3_mkdir_done
smb3_connect_err smb3_mkdir_err
smb3_cred_timeout smb3_nblk_credits
smb3_delete_done smb3_notif_done
smb3_delete_err  smb3_notif_err
smb3_exit_done   smb3_open_done
smb3_exit_err    smb3_open_err
smb3_falloc_done smb3_oplock_not_found
smb3_falloc_err  smb3_overflow_credits
smb3_falloc_err  smb3_partial_send_reconnect
smb3_flush_done  smb3_pend_credits
smb3_flush_enter smb3_posix_mkdir_done
smb3_flush_enter smb3_posix_mkdir_err
smb3_ses_expired smb3_set_credits
smb3_set_eof     smb3_set_eof_done
smb3_set_eof_done smb3_set_eof_enter
smb3_set_eof_err smb3_set_info_compound_done
smb3_set_info_compound_enter smb3_set_info_compound_err
smb3_set_info_err smb3_slow_rsp
smb3_tcon        smb3_tdis_done
smb3_tdis_done   smb3_tdis_err
smb3_tdis_err    smb3_too_many_credits
smb3_wait_credits smb3_write
smb3_write       smb3_xfer
smb3_xfer
smb3
```
eBPF is amazing ...

- See Brendan Gregg’s website
- Also see e.g. https://wiki.samba.org/index.php/LinuxCIFS_troubleshooting
- Can be as simple to do as “trace-cmd record -e cifs”
  - And then “trace-cmd show” in another window
- Let us know if suggestions on other debugging tracepoints that would be helpful
- And don’t forget about proc/fs/cifs/Stats, proc/fs/cifs/open_files and proc/fs/cifs/DebugData ...
Recent improvements – cifs-utils

Userspace tools
Improved user space tools (cifs-utils)

- cifs-utils 7 released in August
  - Add support for gss-proxy (improving krb5 credential retrieval)
  - Misc. bug fixes

- Contributions welcome – lots of cool opportunities for tooling
  - e.g. to leverage the new notify ioctl or to make snapshot mounts easier (one step instead of two) or to improve backup tooling
Coming soon ...

New features under development for SMB3.1.1 on Linux
What features can you expect in next few releases?

- Analyze cases where use of directory leases, deferred close operations could better optimize network traffic while caching safely
- Add use of compounding in more cases or extend it (e.g. open/querydir/querydir instead of open/querydir), and better use existing file leases for compound reqs which include SMB3 open
- Improvements to performance when low on SMB3 credits
- Continued focus on multichannel performance improvements
  - Dynamically adding channels better, and picking optimal channels in special cases
- SMB3.1.1 compression support (allow compressing network based on the SMB3.1.1 compress mount parm)
What features can you expect in next few releases?

- Packet signing performance improvements
- Reenabling support for swapfile over SMB3.1.1 mounts
- Support for creating with O_TMPFILE
- statx to return additional SMB3.1.1 attributes like “offline”
- Improvements to enable fanotify/inotify over SMB3.1.1 mounts (currently requires a private SMB3.1.1 specific ioctl)
- Prototype of SMB3.1.1 over QUIC (new encrypted network transport)
- More perf improvements for folios, cache, parallel i/o, multichannel (thank you Dave Howells, Matthew Wilcox et al)
- More testing of the SMB3.1.1 POSIX with new Samba server
Automated testing has greatly improved

- Historically SMB3.1.1 plugfests multiple times a year have helped.
- The ‘buildbot’ continues to improve, more tests added, reducing regressions and improving quality, migrating to new better hosts.
- Test groups for different server types and a general “cifs-testing” one.
Additional tests are encouraged

- Xfstests are the standard Linux filesystem functional tests
- “Buildbot” is in the process of being upgraded/migrated to new hosts
- Last year added 21 to the main “cifs-testing” regression testing group (up to 245 tests run on every checkin from this group)
- Various server specific groups have added even more
  - Azure SMB3.1.1 multichannel: up 25% more tests, now includes 133 tests
  - Ksmbd (Linux kernel server target) up 15%, now includes 144 tests
- Detailed wiki pages on wiki.samba.org go through how to set up xfstests with cifs.ko, and what features need to be added to include more tests (tests that currently skip or fail so aren’t run in the buildbot)
Thank you for your time

- Future is very bright!
Additional Resources to Explore for SMB3 and Linux

- Samba-technical mailing list and IRC channel
- And various presentations at [http://www.sambaxp.org](http://www.sambaxp.org) and Microsoft channel 9 and of course SNIA … [http://www.snia.org/events/storage-developer](http://www.snia.org/events/storage-developer)
- And the code:
  - [https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/fs/cifs](https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/fs/cifs)
  - For pending changes, soon to go into upstream kernel see:
    - [https://git.samba.org/?p=sfrench/cifs-2.6.git;a=shortlog;h=refs/heads/for-next](https://git.samba.org/?p=sfrench/cifs-2.6.git;a=shortlog;h=refs/heads/for-next)
  - Kernel server code: [https://git.samba.org/ksmbd.git/?p=ksmbd.git](https://git.samba.org/ksmbd.git/?p=ksmbd.git) (ksmbd-for-next branch)