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

Steve French

Azure Storage – Microsoft

Samba Team

and

Jeremy Allison

Google/Samba Team

SMB 3.1.1 POSIX

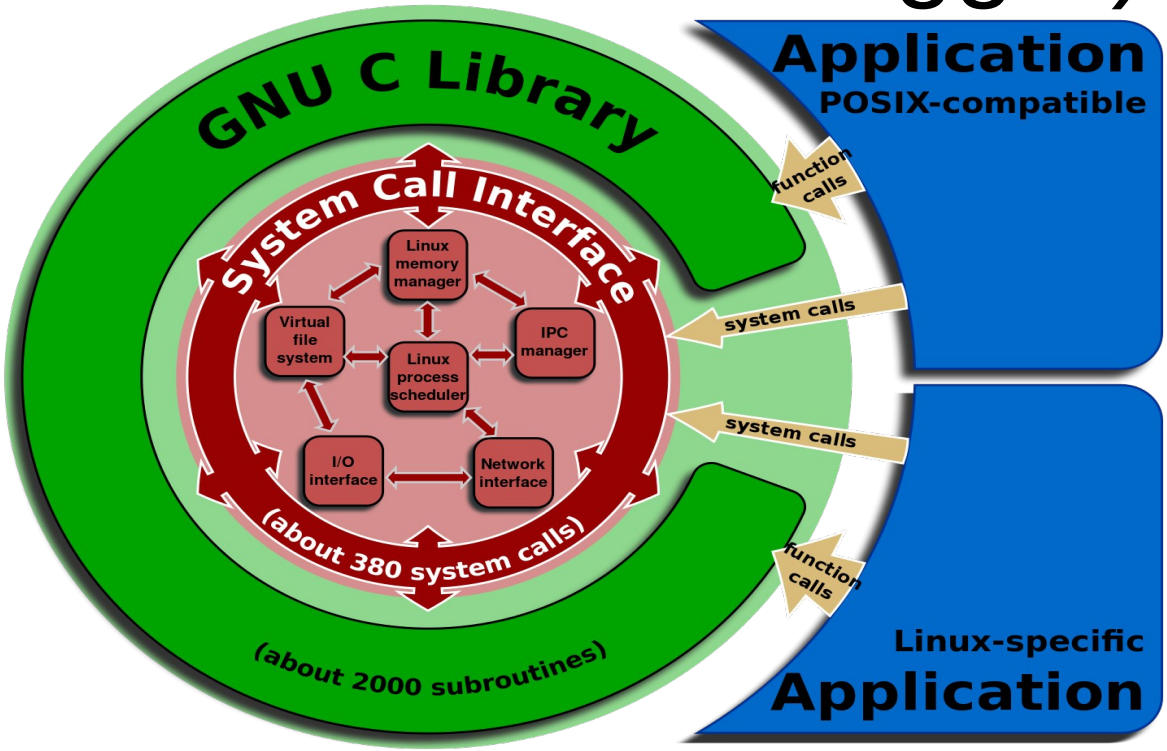
Legal Statement

- This work represents the views of the author(s) and does not necessarily reflect the views of Microsoft or Google
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.

Outline

- What is POSIX?
- Why do these extensions matter?
- Demo
- What if we don't have them? What works?
- Some details
- How to handle Linux continuing to extend APIs, to improve?
- Wireshark and Tracing

POSIX != Linux
(Linux API is much bigger)



Linux is BIG

- Currently > 400 Linux syscalls!

  (try “git grep SYSCALL_DEFINE”

  well over 800 and 400+ are

  even documented “man syscalls”

  FS layer has 222)

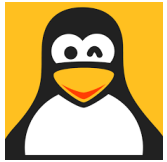
  vs

- Only about 100 POSIX API calls

Example of how Linux is evolving

- Fallocate alone now has 7 flags
  - Insert range
  - Unshare range
  - Zero range
  - Keep size
  - ...
- Rename (renameat2) now has 3 flags
  - noreplace, whiteout and exchange
- And these are just two syscalls ...

Motivations for Extensions



- Linux Apps work!
  - Case sensitivity e.g. is required for the kernel to build on Linux
  - (And Linux and other posix-like operating systems want posix behavior for files whether on premise or in cloud)
- Improve common situations where customers have Linux and Windows and Mac clients accessing the same data
- Deprecation of CIFS – make sure extensions work with most secure, most optimal SMB3.1.1 dialect

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!

Samba.IO.lab
Welcome

A year ago … and now … kernel (including SMB3 client cifs.ko) improving

- A year ago Linux 4.18 “Merciless Moray”

A week ago: 5.3 “Bobtail Squid”

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

Quality Much Improved – Top Priority

- More xfstests pass (up to 127 now and growing), vast majority of the rest are skipped due to missing features or being inappropriate for network file systems
- Crediting (flow control) hugely improved (thanks to Pavel Shilovsky and others)
- Many potential issues pointed out by static analysis addressed
- **The "Buildbot"** reduced regressions and is VERY exciting recent addition for CIT (thanks to 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 (e.g. at the SNIA SMB3 test event in January)

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.

POSIX: Why isn’t it shipped already ?

- For obvious reasons these experimental changes not enabled by default:
  - With current mainline Linux 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 (e.g. at the test event in January)

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

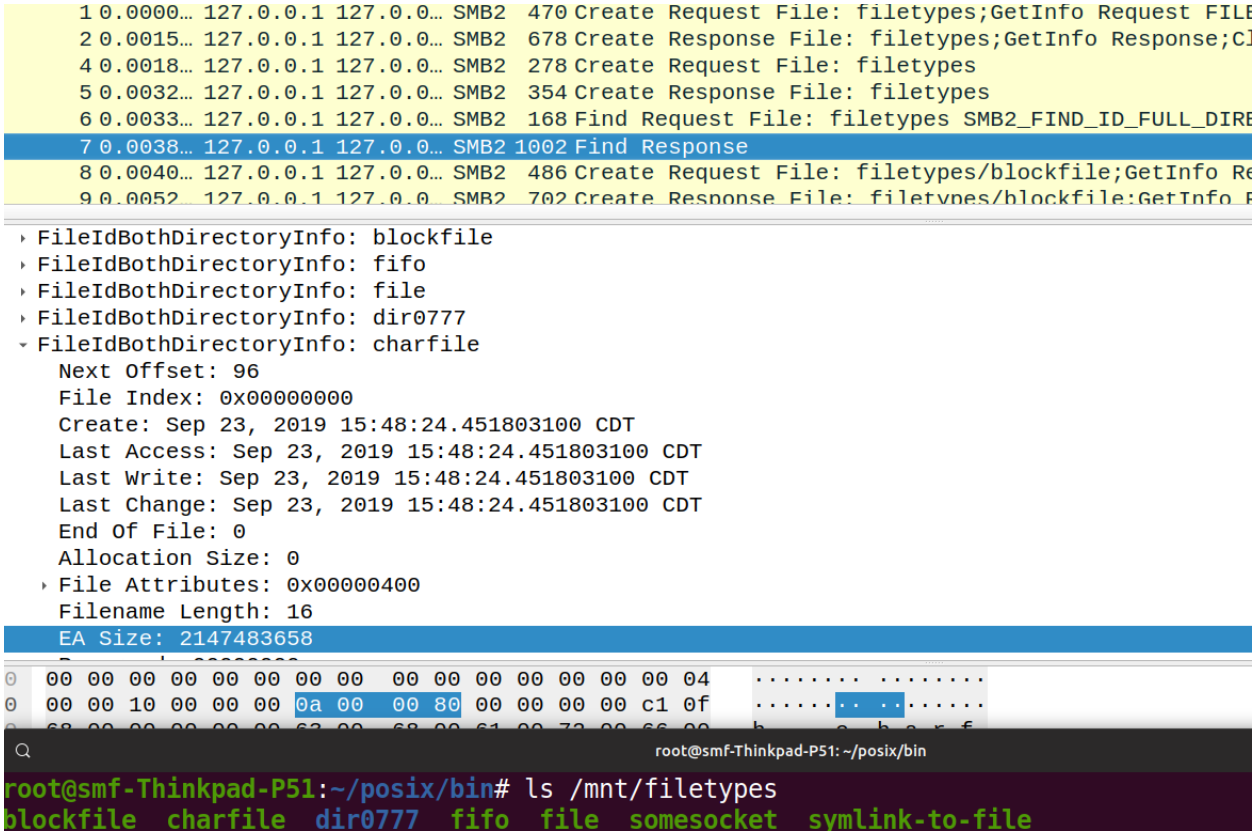
Block special file. Tag value is 0x80000026.

Example: using it today

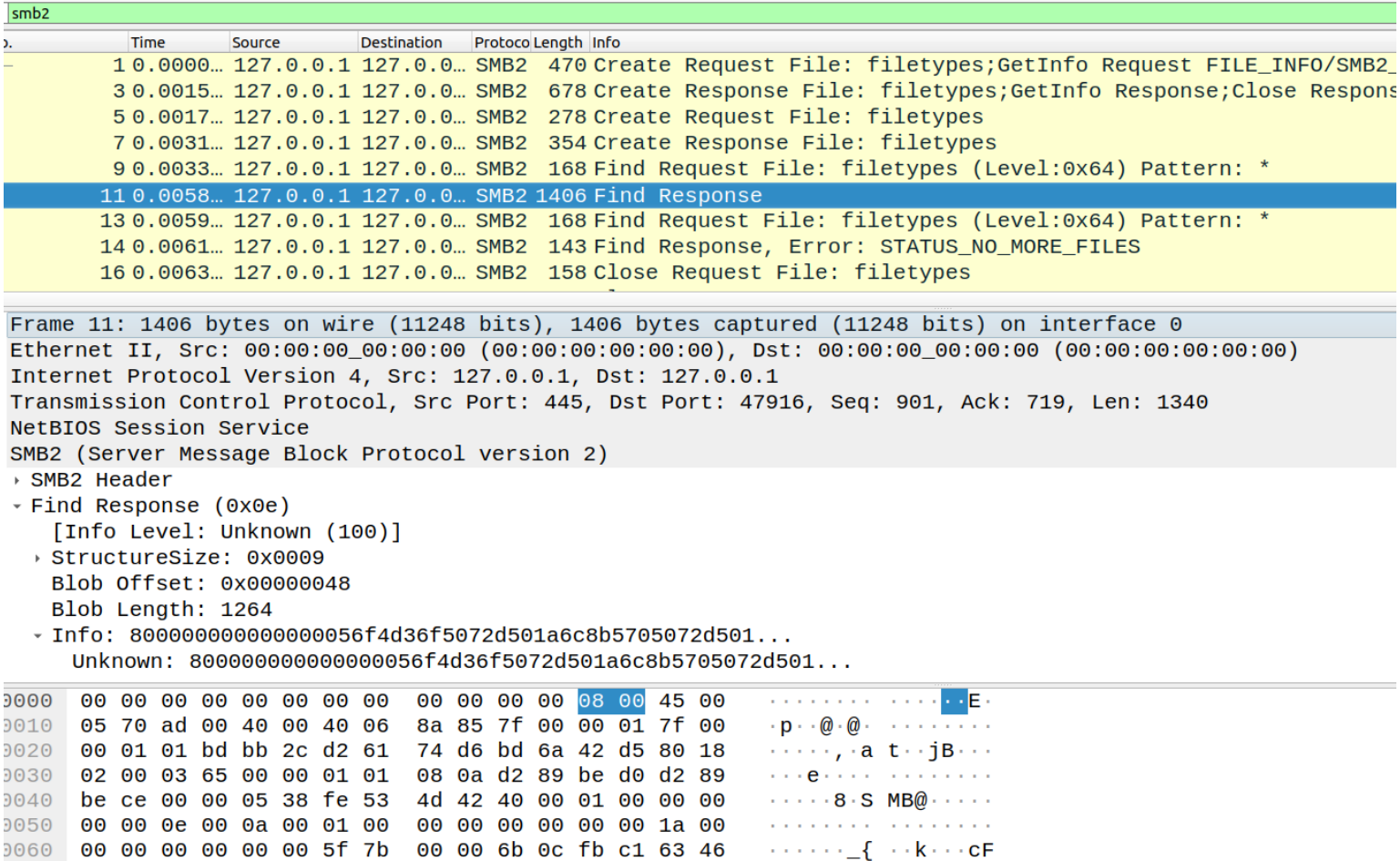
- On the client:
  - “mount –t smb3 //<address>/<share> /mnt -o username=<user>,password=<pass>, vers=3.1.1,posix,mfsymlinks,nomapposix,noperm
- On the server add to smb.conf
  - “mangled names = no”
  - “directory mask = 07777”
  - “create mask = 07777”
  - Consider removing “obey pam restrictions”

Note that directory enumeration returns a reparse point tag (for file type).

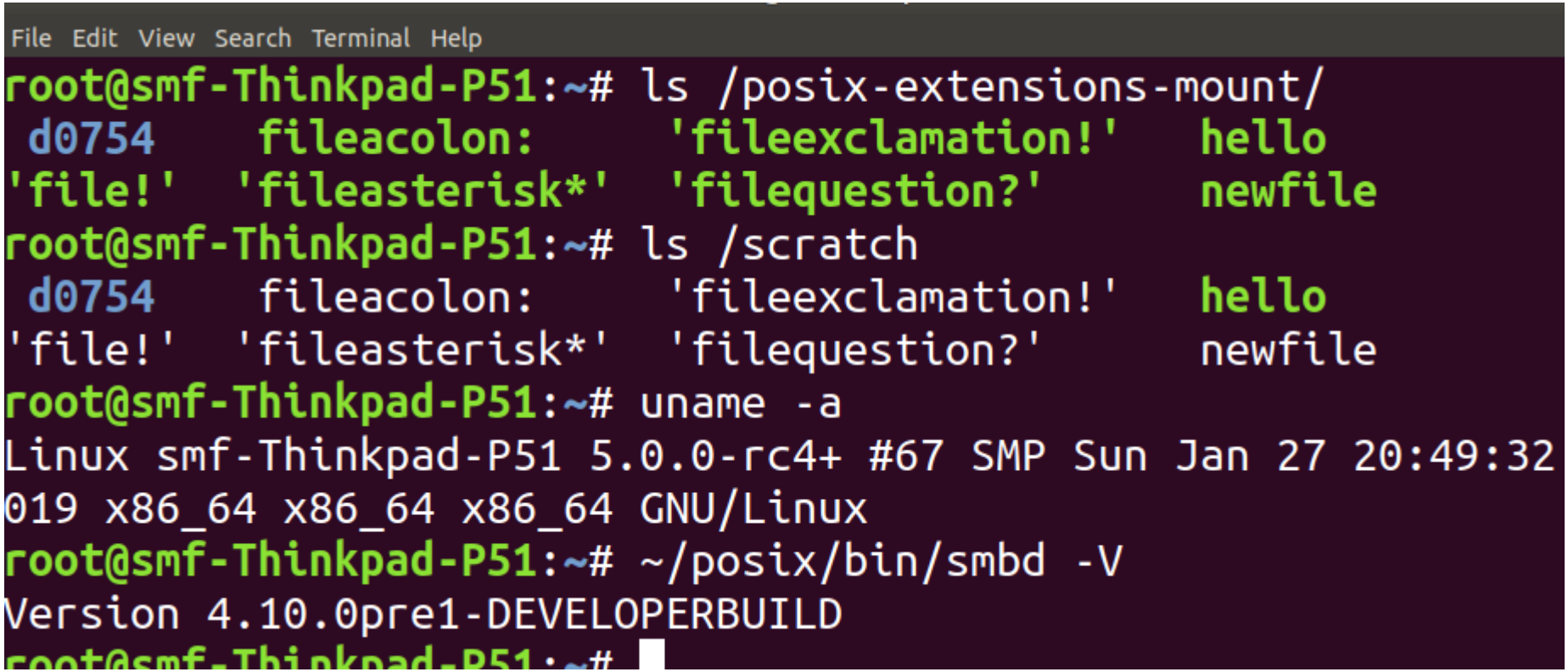
- The proposed change to add tags this would reduce roundtrips



SMB3.1.1 POSIX Query Directory (from today)



POSIX Path Names Work



Note the new mount option “posix” (vs “nounix”)

```
root@Ubuntu-17-Virtual-Machine:~/cifs-2.6# cat /proc/mounts | grep cifs
//localhost/test-no-posix /mnt1 cifs rw,relatime,vers=3.1.1,cache=strict,username=testuser,domain=,uid=0,noforc
euid,gid=0,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,nounix,serverino,mapposix,rsize=1048576,
wsize=1048576,echo_interval=60,actimeo=1 0 0
//localhost/test /mnt cifs rw,relatime,vers=3.1.1,cache=strict,username=testuser,domain=,uid=0,noforceuid,gid=0
,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,posix,posixpaths,serverino,mapposix,rsize=1048576,
wsize=1048576,echo_interval=60,actimeo=1 0 0
root@Ubuntu-17-Virtual-Machine:~/cifs-2.6# cat /proc/fs/cifs/DebugData
Display Internal CIFS Data Structures for Debugging
---------------------------------------------------
CIFS Version 2.12
Features: dfs fscache lanman posix spnego xattr acl
Active VFS Requests: 0
Servers:
Number of credits: 16 Dialect 0x311 posix
1) Name: 127.0.0.1 Uses: 2 Capability: 0x300047 Session Status: 1       TCP status: 1
        Local Users To Server: 1 SecMode: 0x1 Req On Wire: 0
        Shares:
        0) IPC: \\127.0.0.1\IPC$ Mounts: 1 DevInfo: 0x0 Attributes: 0x0
        PathComponentMax: 0 Status: 1 type: 0
        Share Capabilities: None        Share Flags: 0x0
        tid: 0x4f5511db Maximal Access: 0x1f00a9

        1) \\localhost\test Mounts: 1 DevInfo: 0x20 Attributes: 0x1006f
        PathComponentMax: 255 Status: 1 type: DISK
        Share Capabilities: None Aligned, Partition Aligned,    Share Flags: 0x0
        tid: 0x8579c31d Optimal sector size: 0x200      Maximal Access: 0x1f01ff

        2) \\localhost\test-no-posix Mounts: 1 DevInfo: 0x20 Attributes: 0x1006f
        PathComponentMax: 255 Status: 1 type: DISK
        Share Capabilities: None Aligned, Partition Aligned,    Share Flags: 0x0
        tid: 0x1813a493 Optimal sector size: 0x200      Maximal Access: 0x1f01ff

        MIDs:
```

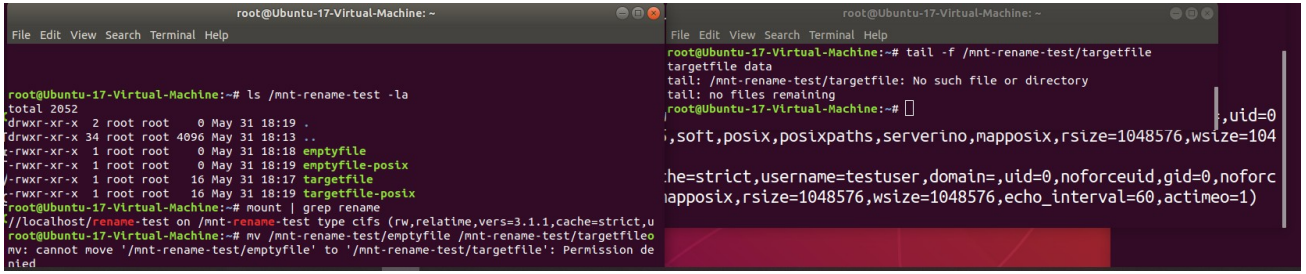
Mode bits on create and case sensitivity work!

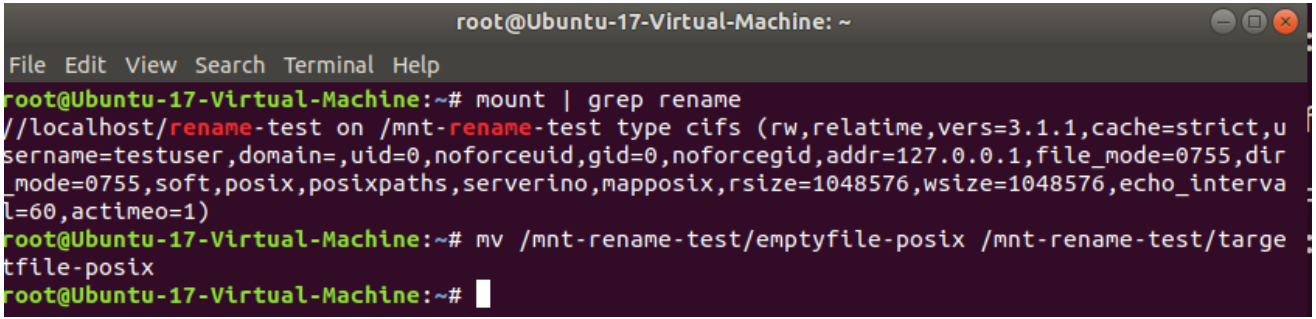
```
root@Ubuntu-17-Virtual-Machine:/mnt# ~/create-4-files-with-mode-test
root@Ubuntu-17-Virtual-Machine:/mnt# cd /mnt1
root@Ubuntu-17-Virtual-Machine:/mnt1# ~/create-4-files-with-mode-test
root@Ubuntu-17-Virtual-Machine:/mnt1# ls /test /test-no-posix -la
/test:
total 12
drwxrwxrwx  3 root     root     4096 May 31 16:55 .
drwxr-xr-x 32 root     root     4096 May 31 16:46 ..
-rwx------  1 testuser testuser    0 May 31 16:55 0700
-rwxrwx---  1 testuser testuser    0 May 31 16:55 0770
-rwxrwxr-x  1 testuser testuser    0 May 31 16:55 0775
drwxr-xr-x  2 sfrench  sfrench  4096 Mar 24 10:34 tmp

/test-no-posix:
total 8
drwxrwxrwx  2 root     root     4096 May 31 16:55 .
drwxr-xr-x 32 root     root     4096 May 31 16:46 ..
-rwxrw-r--  1 testuser testuser    0 May 31 16:55 0700
-rwxrw-r--  1 testuser testuser    0 May 31 16:55 0770
-rwxrw-r--  1 testuser testuser    0 May 31 16:55 0775
root@Ubuntu-17-Virtual-Machine:/mnt1# mkdir UPPER
root@Ubuntu-17-Virtual-Machine:/mnt1# touch upper
root@Ubuntu-17-Virtual-Machine:/mnt1# cd /mnt
root@Ubuntu-17-Virtual-Machine:/mnt# mkdir UPPER
root@Ubuntu-17-Virtual-Machine:/mnt# touch upper
root@Ubuntu-17-Virtual-Machine:/mnt# ls /test /test-no-posix
/test:
0700  0770  0775  tmp  upper  UPPER

/test-no-posix:
0700  0770  0775  UPPER
```

Mode bits on mkdir works!

```
root@smf-Thinkpad-P51:~/cifs-2.6# mount -t smb3 //127.0.0.1/scratch /mnt -o username=testuser,
,vers=3.11,posix
root@smf-Thinkpad-P51:~/cifs-2.6# umask 0000
root@smf-Thinkpad-P51:~/cifs-2.6# mkdir /mnt/0774 -m 0774
root@smf-Thinkpad-P51:~/cifs-2.6# mkdir /mnt/0770 -m 0770
root@smf-Thinkpad-P51:~/cifs-2.6# mkdir /mnt/0444 -m 0444
root@smf-Thinkpad-P51:~/cifs-2.6# ls /scratch -la
total 20
drwxrwxrwx  5 root     root     4096 Jun 15 20:42 .
drwxr-xr-x 35 root     root     4096 Jun 15 20:39 ..
dr--r--r--  2 testuser testuser 4096 Jun 15 20:42 0444
drwxrwx---  2 testuser testuser 4096 Jun 15 20:42 0770
drwxrwxr--  2 testuser testuser 4096 Jun 15 20:42 0774
root@smf-Thinkpad-P51:~/cifs-2.6# cat /proc/version
Linux version 4.17.0+ (sfrench@smf-Thinkpad-P51) (gcc version 7.3.0 (Ubuntu 7.3.0-16ubuntu3))
```

Rename works with POSIX extensions!





Statfs (“stat –f”) without POSIX extensions:

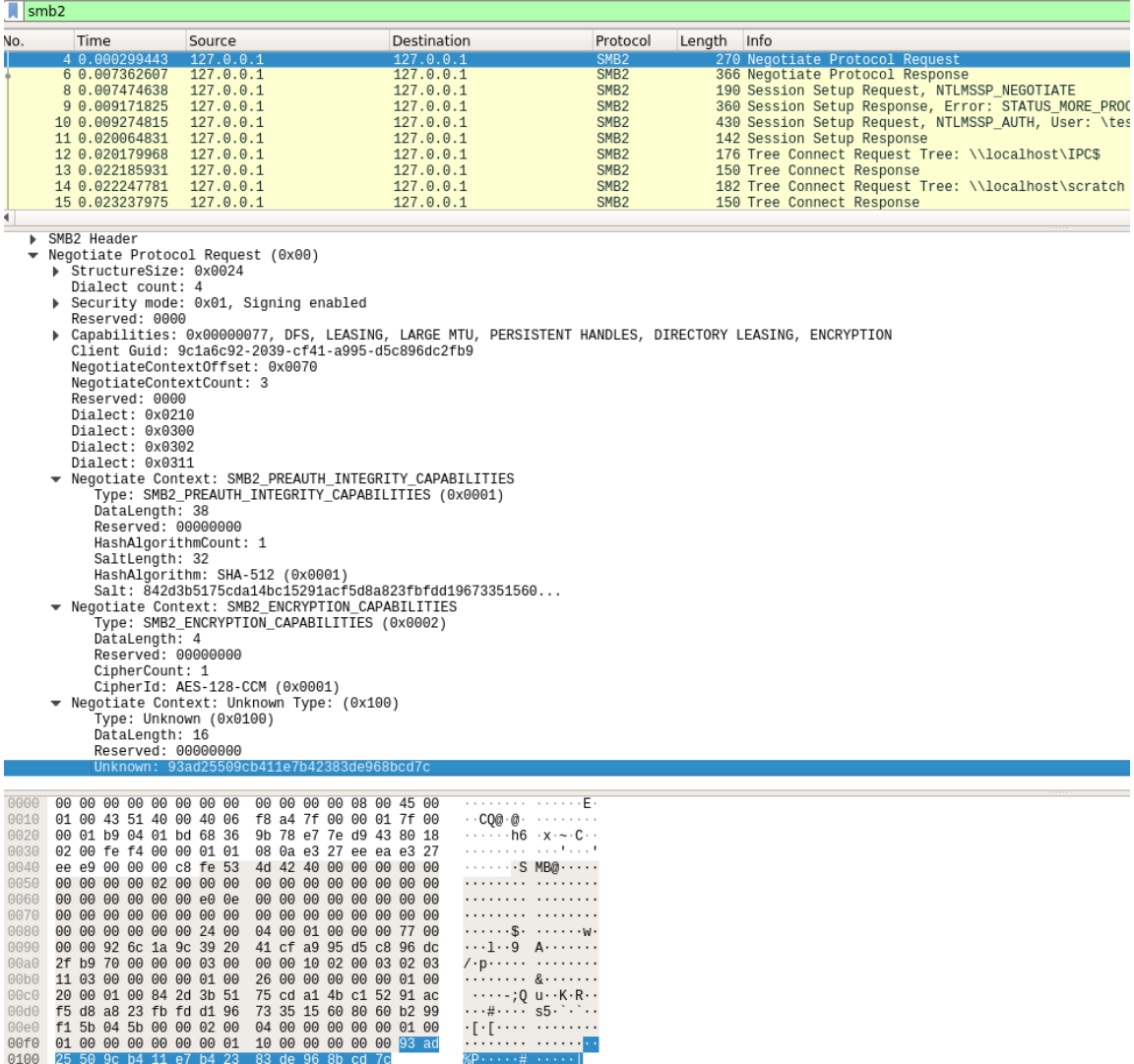
```
root@smf-Thinkpad-P51:~/cifs-2.6# cat /proc/mounts | grep cifs
//localhost/scratch /mnt cifs rw,relatime,vers=3.0,cache=strict,username=testuser,domai
,uid=0,noforceuid,gid=0,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,nou
x,serverino,mfsymlinks,noperm,rsize=1048576,wsize=1048576,echo_interval=60,actimeo=1 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!

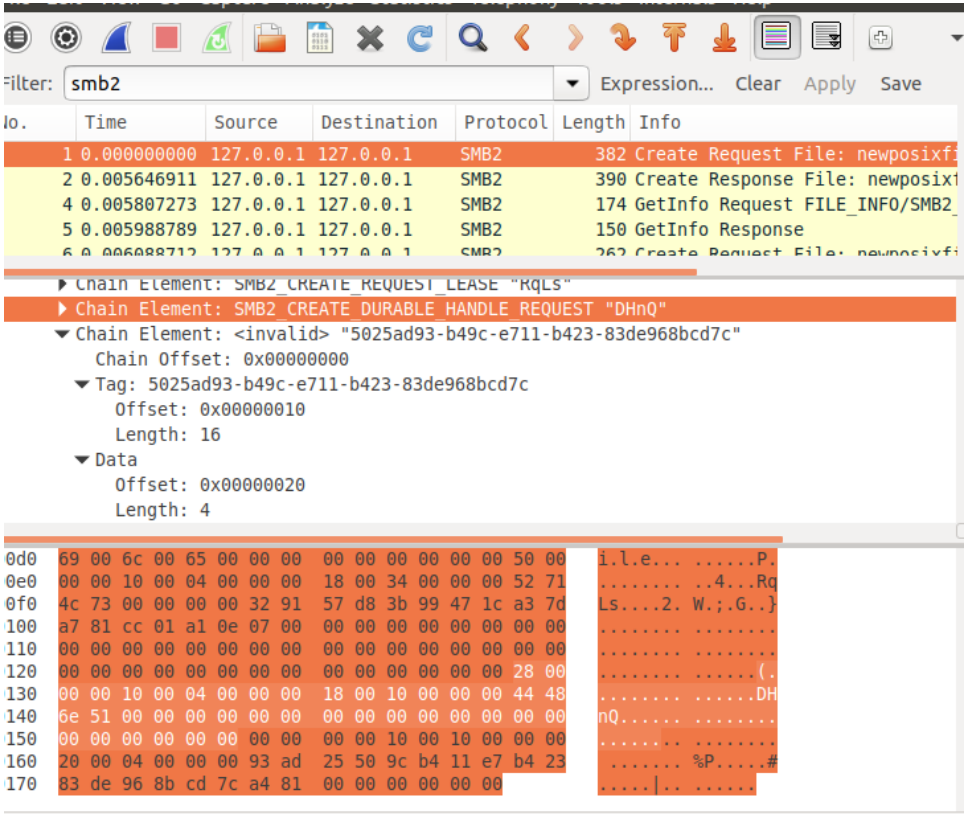
```
root@smf-Thinkpad-P51:~/cifs-2.6# cat /proc/mounts | grep smb3
//127.0.0.1/scratch /mnt1 smb3 rw,relatime,vers=3.1.1,cache=strict,username=testuser,doma
in=,uid=0,noforceuid,gid=0,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,po
six,posixpaths,serverino,mapposix,noperm,rsize=1048576,wsize=1048576,echo_interval=60,act
imeo=1 0 0
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: 13901536
```

Details – Negotiate Req (w/POSIX)

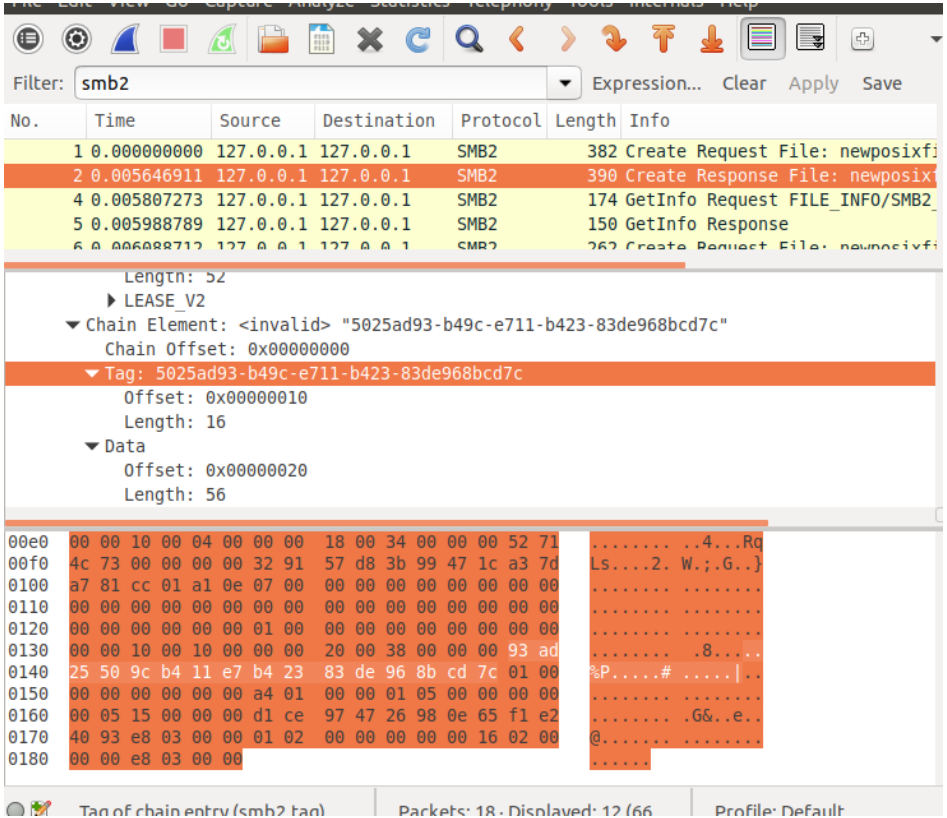
- Format has changed since 2018 SDC
- Now 16 byte GUID
- Allows versioning



Details continued – Create (POSIX) req



Details continued – create response



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
```

CIFS Unix/POSIX Extensions

- What was wrong with what we had?
  - Remember CIFS Deprecation?
  - And not just due to WannaCry …
    - SMB3 is really good …
- Apple SMB2/SMB3 create context does handle case sensitivity, but not all POSIX compatibility issues

Client Perspective

- 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

The challenges of Create/ Rename/Delete

The challenges of POSIX inode metadata

- What do we need to be able to return?
- What about mode bits and ACLs?

The Challenges of POSIX locking

The Challenges of POSIX FS info

Remember the Server Perspective?

Details

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-description-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

POSIX Extensions – Where do we go from here?

- Continue debugging test implementations (cifs.ko and JRAs Samba POSIX test branch). Current focus (eg at this test event) on enhancing smb3 client to handle mode bits in ACL, and handle special files and server side symlinks via reparse points
- 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.) to be stored as reparse points enables five tests.
- Continue extending the wireshark dissectors (see Aurelien)
- Continue testing here, and in two weeks in Redmond and in September at SNIA SDC
- 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 ...

+ ***S M B 3***