How Compliant is the Linux client?

Steve French

Azure Storage – Microsoft

Samba Team

SMB 3.1.1

Legal Statement

Outline

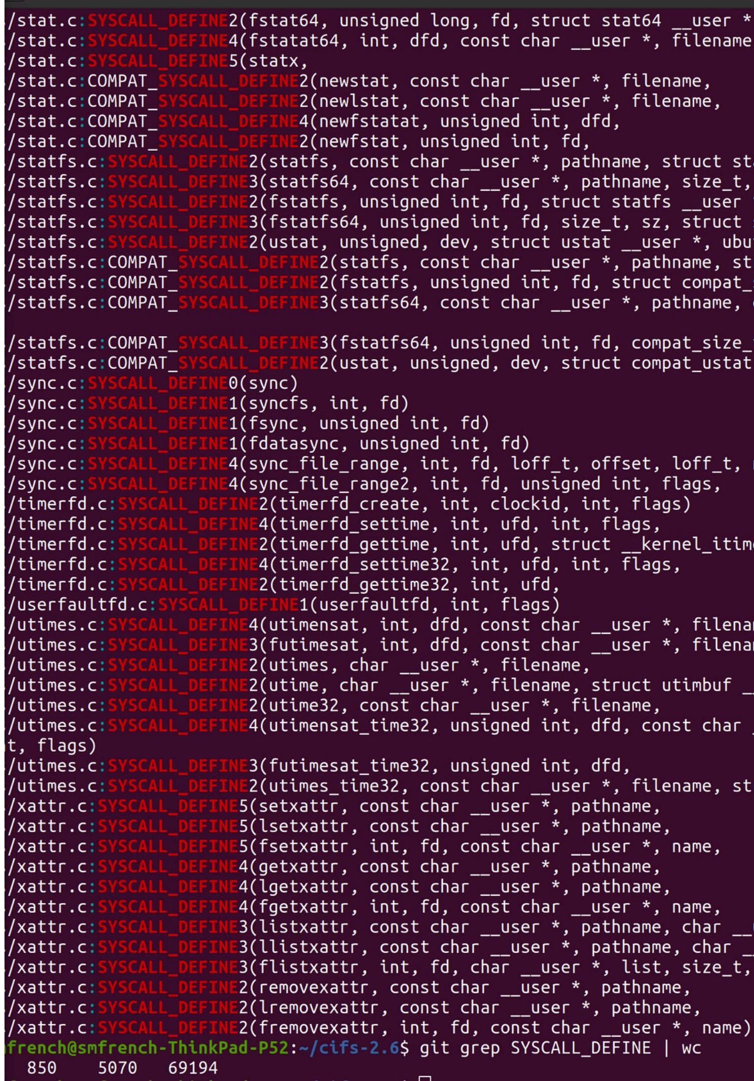
- Linux is a lot more than POSIX ...
- What works today?
    - Without POSIX Extensions
    - With POSIX extensions
- Some xfstesting details
- What to work on next?
- How to handle Linux continuing to extend APIs and test it ...

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



```
/stat.c:SYSCALL_DEFINE2(fstat64, unsigned long, fd, struct stat64 __user *
/stat.c:SYSCALL_DEFINE4(fstatat64, int, dfd, const char __user *, filename
/stat.c:SYSCALL_DEFINE5(statx,
/stat.c:COMPAT_SYSCALL_DEFINE2(newstat, const char __user *, filename,
/stat.c:COMPAT_SYSCALL_DEFINE2(newlstat, const char __user *, filename,
/stat.c:COMPAT_SYSCALL_DEFINE4(newfstatat, unsigned int, dfd,
/stat.c:COMPAT_SYSCALL_DEFINE2(newfstat, unsigned int, fd,
/statfs.c:SYSCALL_DEFINE2(statfs, const char __user *, pathname, struct st
/statfs.c:SYSCALL_DEFINE3(statfs64, const char __user *, pathname, size_t,
/statfs.c:SYSCALL_DEFINE2(fstatfs, unsigned int, fd, struct statfs __user
/statfs.c:SYSCALL_DEFINE3(fstatfs64, unsigned int, fd, size_t, sz, struct
/statfs.c:SYSCALL_DEFINE2(ustat, unsigned, dev, struct ustat __user *, ubu
/statfs.c:COMPAT_SYSCALL_DEFINE2(statfs, const char __user *, pathname, st
/statfs.c:COMPAT_SYSCALL_DEFINE2(fstatfs, unsigned int, fd, struct compat_
/statfs.c:COMPAT_SYSCALL_DEFINE3(statfs64, const char __user *, pathname,

/statfs.c:COMPAT_SYSCALL_DEFINE3(fstatfs64, unsigned int, fd, compat_size_
/statfs.c:COMPAT_SYSCALL_DEFINE2(ustat, unsigned, dev, struct compat_ustat
/sync.c:SYSCALL_DEFINE0(sync)
/sync.c:SYSCALL_DEFINE1(syncfs, int, fd)
/sync.c:SYSCALL_DEFINE1(fsync, unsigned int, fd)
/sync.c:SYSCALL_DEFINE1(fdatasync, unsigned int, fd)
/sync.c:SYSCALL_DEFINE4(sync_file_range, int, fd, loff_t, offset, loff_t,
/sync.c:SYSCALL_DEFINE4(sync_file_range2, int, fd, unsigned int, flags,
/timerfd.c:SYSCALL_DEFINE2(timerfd_create, int, clockid, int, flags)
/timerfd.c:SYSCALL_DEFINE4(timerfd_settime, int, ufd, int, flags,
/timerfd.c:SYSCALL_DEFINE2(timerfd_gettime, int, ufd, struct __kernel_itim
/timerfd.c:SYSCALL_DEFINE4(timerfd_settime32, int, ufd, int, flags,
/timerfd.c:SYSCALL_DEFINE2(timerfd_gettime32, int, ufd,
/userfaultfd.c:SYSCALL_DEFINE1(userfaultfd, int, flags)
/utimes.c:SYSCALL_DEFINE4(utimensat, int, dfd, const char __user *, filena
/utimes.c:SYSCALL_DEFINE3(futimesat, int, dfd, const char __user *, filena
/utimes.c:SYSCALL_DEFINE2(utimes, char __user *, filename,
/utimes.c:SYSCALL_DEFINE2(utime, char __user *, filename, struct utimbuf _
/utimes.c:SYSCALL_DEFINE2(utime32, const char __user *, filename,
/utimes.c:SYSCALL_DEFINE4(utimensat_time32, unsigned int, dfd, const char
t, flags)
/utimes.c:SYSCALL_DEFINE3(futimesat_time32, unsigned int, dfd,
/utimes.c:SYSCALL_DEFINE2(utimes_time32, const char __user *, filename, st
/xattr.c:SYSCALL_DEFINE5(setxattr, const char __user *, pathname,
/xattr.c:SYSCALL_DEFINE5(lsetxattr, const char __user *, pathname,
/xattr.c:SYSCALL_DEFINE5(fsetxattr, int, fd, const char __user *, name,
/xattr.c:SYSCALL_DEFINE4(getxattr, const char __user *, pathname,
/xattr.c:SYSCALL_DEFINE4(lgetxattr, const char __user *, pathname,
/xattr.c:SYSCALL_DEFINE4(fgetxattr, int, fd, const char __user *, name,
/xattr.c:SYSCALL_DEFINE3(listxattr, const char __user *, pathname, char __
/xattr.c:SYSCALL_DEFINE3(llistxattr, const char __user *, pathname, char _
/xattr.c:SYSCALL_DEFINE3(flistxattr, int, fd, char __user *, list, size_t,
/xattr.c:SYSCALL_DEFINE2(removexattr, const char __user *, pathname,
/xattr.c:SYSCALL_DEFINE2(lremovexattr, const char __user *, pathname,
/xattr.c:SYSCALL_DEFINE2(fremovexattr, int, fd, const char __user *, name)
french@smfrench-ThinkPad-P52:~/cifs-2.6$ git grep SYSCALL_DEFINE | wc
    850    5070   69194
```

Linux filesystems are not easy! Responsible for more than 200 of 850 syscalls. +4 since last year

Syscall name	Kernel Version introduced
epoll_pwait2	5.11
mount_setattr	5.12
faccessat2	5.8
close_range	5.9

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

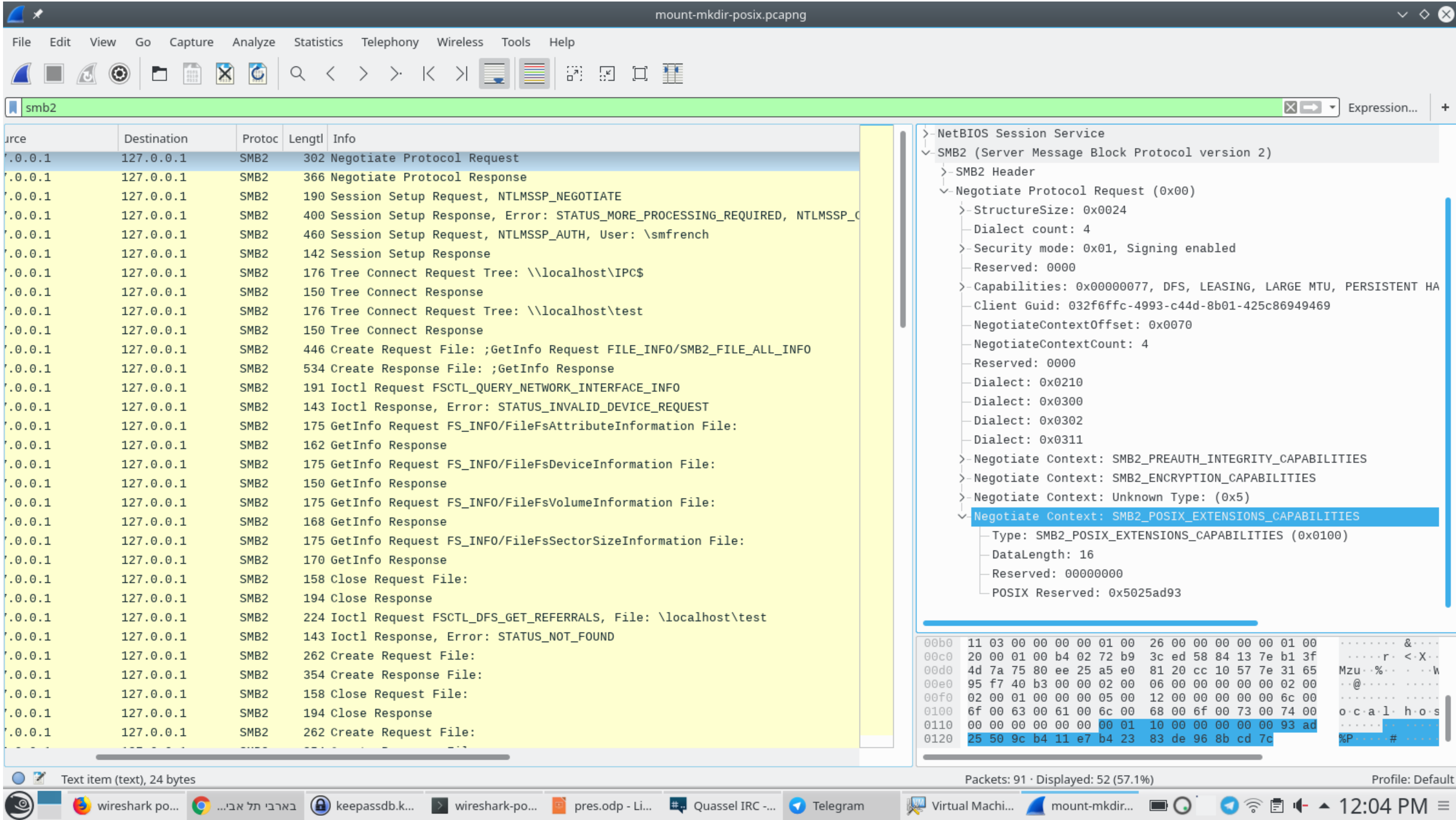
- Rename over an open file
- Files with pending delete showing up in the namespace (e.g. readdir)
- Case sensitive file names
- Locking scenarios that require advisory locks
- Populating a few fields in the statfs response (e.g. total and free inodes)
- Ownership can be preserved with “idsfromsid” on 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


mount-mkdir-posix.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

smb2 Expression...

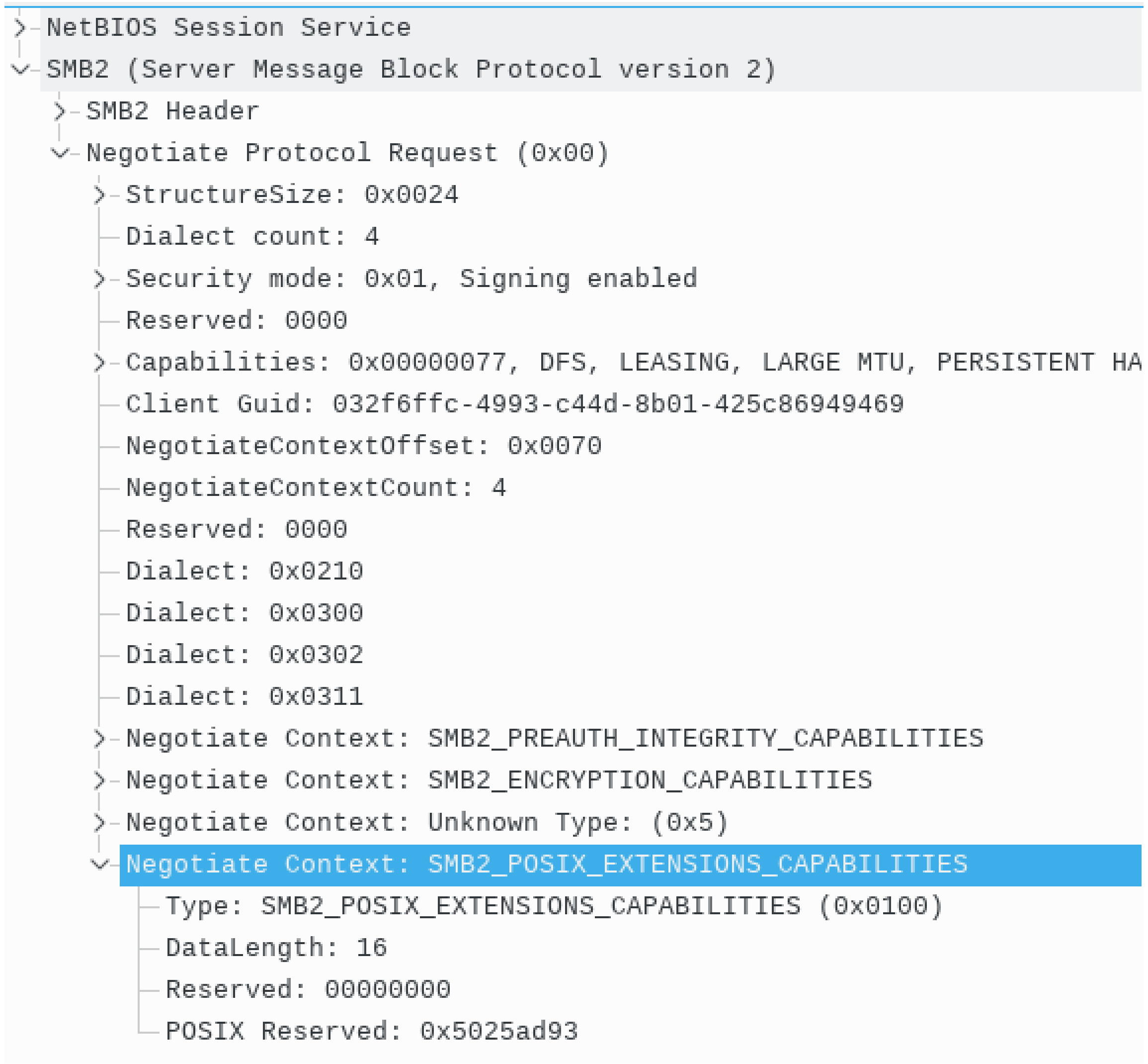
```
urce       Destination   Protoc Lengtl Info
.0.0.1     127.0.0.1     SMB2    302 Negotiate Protocol Request
.0.0.1     127.0.0.1     SMB2    366 Negotiate Protocol Response
.0.0.1     127.0.0.1     SMB2    190 Session Setup Request, NTLMSSP_NEGOTIATE
.0.0.1     127.0.0.1     SMB2    400 Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_C
.0.0.1     127.0.0.1     SMB2    460 Session Setup Request, NTLMSSP_AUTH, User: \smfrench
.0.0.1     127.0.0.1     SMB2    142 Session Setup Response
.0.0.1     127.0.0.1     SMB2    176 Tree Connect Request Tree: \\localhost\IPC$
.0.0.1     127.0.0.1     SMB2    150 Tree Connect Response
.0.0.1     127.0.0.1     SMB2    176 Tree Connect Request Tree: \\localhost\test
.0.0.1     127.0.0.1     SMB2    150 Tree Connect Response
.0.0.1     127.0.0.1     SMB2    446 Create Request File: ;GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO
.0.0.1     127.0.0.1     SMB2    534 Create Response File: ;GetInfo Response
.0.0.1     127.0.0.1     SMB2    191 Ioctl Request FSCTL_QUERY_NETWORK_INTERFACE_INFO
.0.0.1     127.0.0.1     SMB2    143 Ioctl Response, Error: STATUS_INVALID_DEVICE_REQUEST
.0.0.1     127.0.0.1     SMB2    175 GetInfo Request FS_INFO/FileFsAttributeInformation File:
.0.0.1     127.0.0.1     SMB2    162 GetInfo Response
.0.0.1     127.0.0.1     SMB2    175 GetInfo Request FS_INFO/FileFsDeviceInformation File:
.0.0.1     127.0.0.1     SMB2    150 GetInfo Response
.0.0.1     127.0.0.1     SMB2    175 GetInfo Request FS_INFO/FileFsVolumeInformation File:
.0.0.1     127.0.0.1     SMB2    168 GetInfo Response
.0.0.1     127.0.0.1     SMB2    175 GetInfo Request FS_INFO/FileFsSectorSizeInformation File:
.0.0.1     127.0.0.1     SMB2    170 GetInfo Response
.0.0.1     127.0.0.1     SMB2    158 Close Request File:
.0.0.1     127.0.0.1     SMB2    194 Close Response
.0.0.1     127.0.0.1     SMB2    224 Ioctl Request FSCTL_DFS_GET_REFERRALS, File: \localhost\test
.0.0.1     127.0.0.1     SMB2    143 Ioctl Response, Error: STATUS_NOT_FOUND
.0.0.1     127.0.0.1     SMB2    262 Create Request File:
.0.0.1     127.0.0.1     SMB2    354 Create Response File:
.0.0.1     127.0.0.1     SMB2    158 Close Request File:
.0.0.1     127.0.0.1     SMB2    194 Close Response
.0.0.1     127.0.0.1     SMB2    262 Create Request File:
```

```
> NetBIOS Session Service
v SMB2 (Server Message Block Protocol version 2)
  > SMB2 Header
  v 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_CAPABILITIES
    > Negotiate Context: SMB2_ENCRYPTION_CAPABILITIES
    > Negotiate Context: Unknown Type: (0x5)
    v Negotiate Context: SMB2_POSIX_EXTENSIONS_CAPABILITIES
        Type: SMB2_POSIX_EXTENSIONS_CAPABILITIES (0x0100)
        DataLength: 16
        Reserved: 00000000
        POSIX Reserved: 0x5025ad93
```

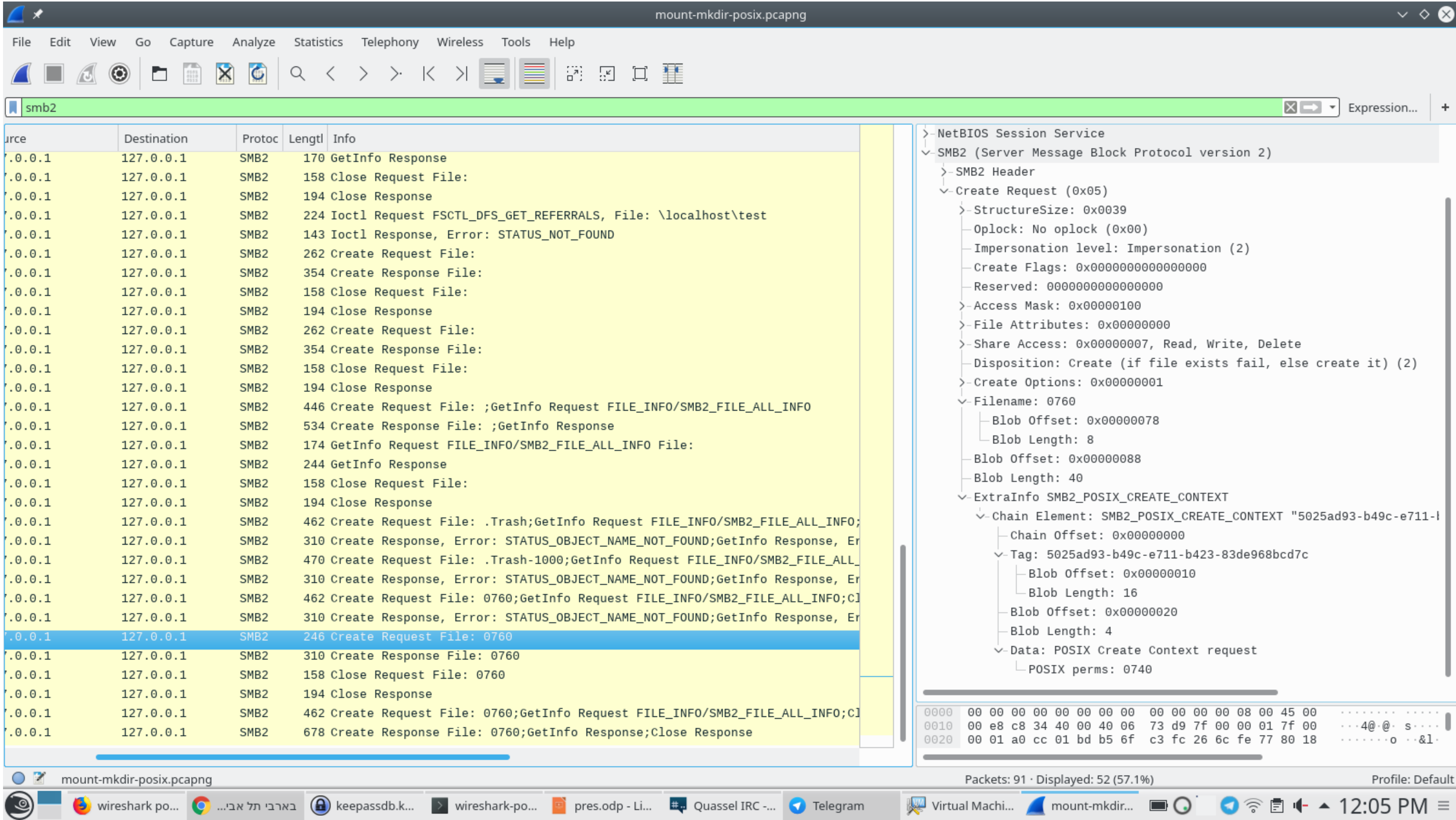
```
00b0  11 03 00 00 00 00 01 00  26 00 00 00 00 00 01 00   ........&.......
00c0  20 00 01 00 b4 02 72 b9  3c ed 58 84 13 7e b1 3f    .....r.<.X..~.?
00d0  4d 7a 75 80 ee 25 a5 e0  81 20 cc 10 57 7e 31 65   Mzu..%... ..W~1e
00e0  95 f7 40 b3 00 00 02 00  06 00 00 00 00 00 02 00   ..@.............
00f0  02 00 01 00 00 00 05 00  12 00 00 00 00 00 6c 00   ..............l.
0100  6f 00 63 00 61 00 6c 00  68 00 6f 00 73 00 74 00   o.c.a.l.h.o.s.t.
0110  00 00 00 00 00 00 00 01  10 00 00 00 00 00 93 ad   ................
0120  25 50 9c b4 11 e7 b4 23  83 de 96 8b cd 7c         %P.....#.....|
```

Text item (text), 24 bytes   Packets: 91 · Displayed: 52 (57.1%)   Profile: Default

wireshark po... | בארבי תל אבי... | keepassdb.k... | wireshark-po... | pres.odp - Li... | Quassel IRC -... | Telegram | Virtual Machi... | mount-mkdir... | 12:04 PM



```
> NetBIOS Session Service
v SMB2 (Server Message Block Protocol version 2)
  > SMB2 Header
  v 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_CAPABILITIES
    > Negotiate Context: SMB2_ENCRYPTION_CAPABILITIES
    > Negotiate Context: Unknown Type: (0x5)
    v Negotiate Context: SMB2_POSIX_EXTENSIONS_CAPABILITIES
        Type: SMB2_POSIX_EXTENSIONS_CAPABILITIES (0x0100)
        DataLength: 16
        Reserved: 00000000
        POSIX Reserved: 0x5025ad93
```

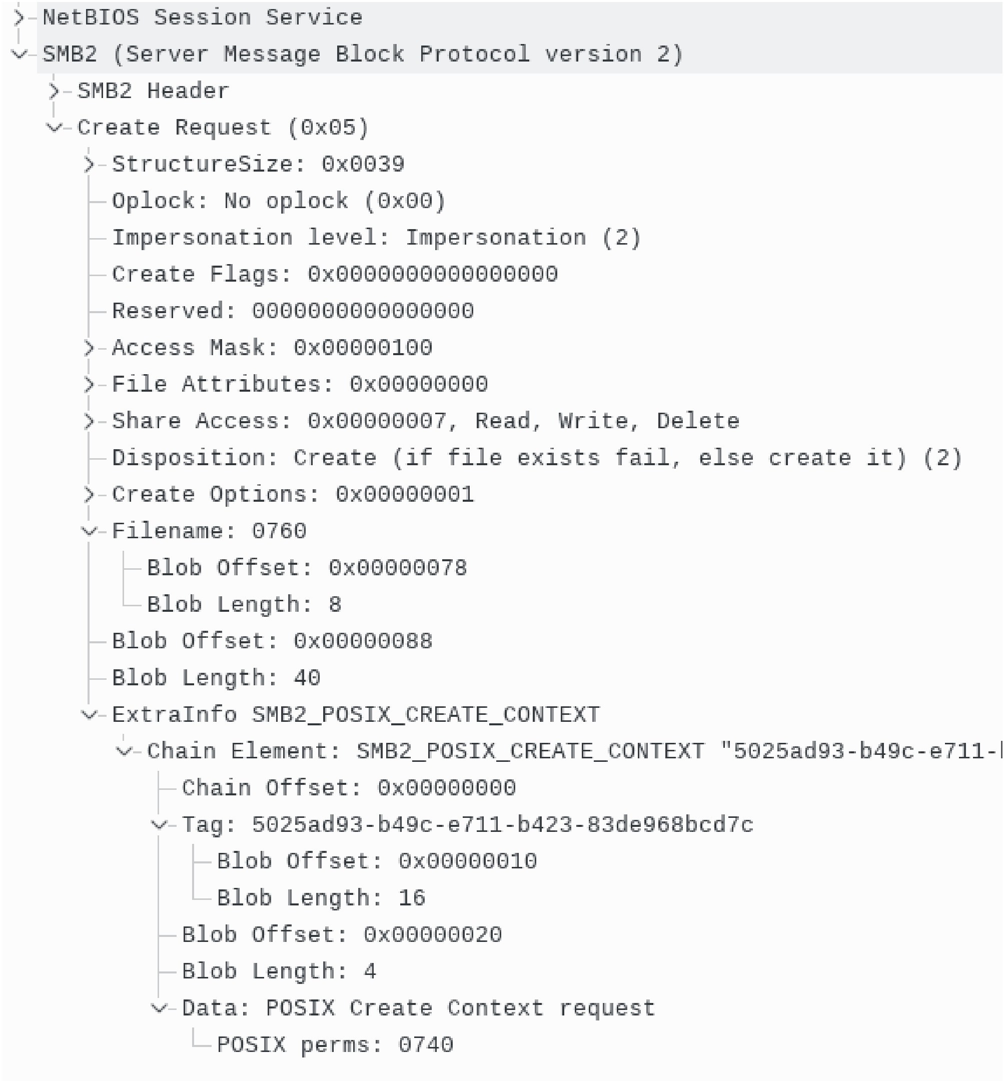

mount-mkdir-posix.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
smb2
Expression...

```
urce      Destination  Protoc Lengtl Info
.0.0.1    127.0.0.1    SMB2   170 GetInfo Response
.0.0.1    127.0.0.1    SMB2   158 Close Request File:
.0.0.1    127.0.0.1    SMB2   194 Close Response
.0.0.1    127.0.0.1    SMB2   224 Ioctl Request FSCTL_DFS_GET_REFERRALS, File: \localhost\test
.0.0.1    127.0.0.1    SMB2   143 Ioctl Response, Error: STATUS_NOT_FOUND
.0.0.1    127.0.0.1    SMB2   262 Create Request File:
.0.0.1    127.0.0.1    SMB2   354 Create Response File:
.0.0.1    127.0.0.1    SMB2   158 Close Request File:
.0.0.1    127.0.0.1    SMB2   194 Close Response
.0.0.1    127.0.0.1    SMB2   262 Create Request File:
.0.0.1    127.0.0.1    SMB2   354 Create Response File:
.0.0.1    127.0.0.1    SMB2   158 Close Request File:
.0.0.1    127.0.0.1    SMB2   194 Close Response
.0.0.1    127.0.0.1    SMB2   446 Create Request File: ;GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO
.0.0.1    127.0.0.1    SMB2   534 Create Response File: ;GetInfo Response
.0.0.1    127.0.0.1    SMB2   174 GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO File:
.0.0.1    127.0.0.1    SMB2   244 GetInfo Response
.0.0.1    127.0.0.1    SMB2   158 Close Request File:
.0.0.1    127.0.0.1    SMB2   194 Close Response
.0.0.1    127.0.0.1    SMB2   462 Create Request File: .Trash;GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO;
.0.0.1    127.0.0.1    SMB2   310 Create Response, Error: STATUS_OBJECT_NAME_NOT_FOUND;GetInfo Response, Er
.0.0.1    127.0.0.1    SMB2   470 Create Request File: .Trash-1000;GetInfo Request FILE_INFO/SMB2_FILE_ALL_
.0.0.1    127.0.0.1    SMB2   310 Create Response, Error: STATUS_OBJECT_NAME_NOT_FOUND;GetInfo Response, Er
.0.0.1    127.0.0.1    SMB2   462 Create Request File: 0760;GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO;Cl
.0.0.1    127.0.0.1    SMB2   310 Create Response, Error: STATUS_OBJECT_NAME_NOT_FOUND;GetInfo Response, Er
.0.0.1    127.0.0.1    SMB2   246 Create Request File: 0760
.0.0.1    127.0.0.1    SMB2   310 Create Response File: 0760
.0.0.1    127.0.0.1    SMB2   158 Close Request File: 0760
.0.0.1    127.0.0.1    SMB2   194 Close Response
.0.0.1    127.0.0.1    SMB2   462 Create Request File: 0760;GetInfo Request FILE_INFO/SMB2_FILE_ALL_INFO;Cl
.0.0.1    127.0.0.1    SMB2   678 Create Response File: 0760;GetInfo Response;Close Response
```

```
> NetBIOS Session Service
v SMB2 (Server Message Block Protocol version 2)
  > SMB2 Header
  v 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
    v Filename: 0760
        Blob Offset: 0x00000078
        Blob Length: 8
      Blob Offset: 0x00000088
      Blob Length: 40
    v ExtraInfo SMB2_POSIX_CREATE_CONTEXT
      v Chain Element: SMB2_POSIX_CREATE_CONTEXT "5025ad93-b49c-e711-l
          Chain Offset: 0x00000000
        v Tag: 5025ad93-b49c-e711-b423-83de968bcd7c
            Blob Offset: 0x00000010
            Blob Length: 16
          Blob Offset: 0x00000020
          Blob Length: 4
        v Data: POSIX Create Context request
            POSIX perms: 0740
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ........ ......E.
0010  00 e8 c8 34 40 00 40 06  73 d9 7f 00 00 01 7f 00   ...4@.@. s.......
0020  00 01 a0 cc 01 bd b5 6f  c3 fc 26 6c fe 77 80 18   .......o ..&l.w..
```

mount-mkdir-posix.pcapng   Packets: 91 · Displayed: 52 (57.1%)   Profile: Default



```
> NetBIOS Session Service
v SMB2 (Server Message Block Protocol version 2)
  > SMB2 Header
  v 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
    v Filename: 0760
        Blob Offset: 0x00000078
        Blob Length: 8
      Blob Offset: 0x00000088
      Blob Length: 40
    v ExtraInfo SMB2_POSIX_CREATE_CONTEXT
      v Chain Element: SMB2_POSIX_CREATE_CONTEXT "5025ad93-b49c-e711-
          Chain Offset: 0x00000000
        v Tag: 5025ad93-b49c-e711-b423-83de968bcd7c
            Blob Offset: 0x00000010
            Blob Length: 16
          Blob Offset: 0x00000020
          Blob Length: 4
        v Data: POSIX Create Context request
            POSIX perms: 0740
```

Some key problems with or without POSIX Extensions

- O_**T**MPFILE support
- Mapping of POSIX ACLs and RichACL on the wire
- SELinux integration
- Case sensitive xattrs
- Better integration of *Q*uota 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 REARRAN***G***E_FILE and **SHUF**FLE_FILE could help if available over SMB3
  - What about exposing Windows's FILE_FLA***G***_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/

More details (continued)

- 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)

More details (continued)

- 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)

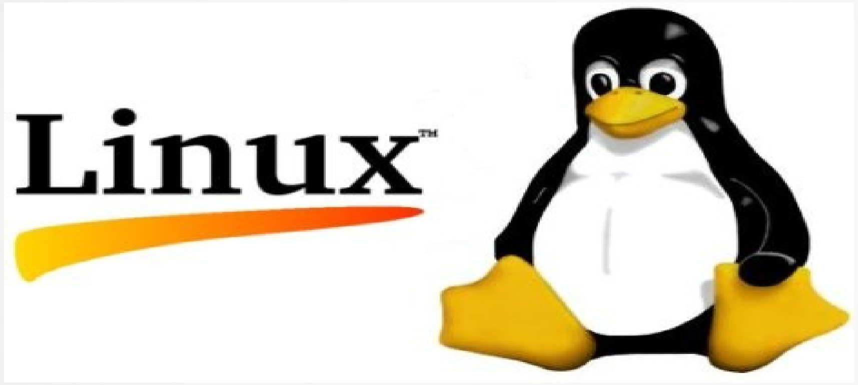
- Conflicting xattrs (test 454)
- XATTR_REPLACE (test 486)
- xfs_io label (492)
- Lsattr -d (508)
- Xattrs with slashes in name (523)
- Casefolding support (556)
- Dupremove utility (559), actton utility (596)
- Fsverity (571)

Next Steps

- Remember we can prototype to ksmbd as well now … and experiment ...

Thank you for your time

- A very exciting time for ...



+ *SMB3*