





**+**

***S***  
***M***  
***B***  
***3.***  
***1***  
***1***

# Legal Statement

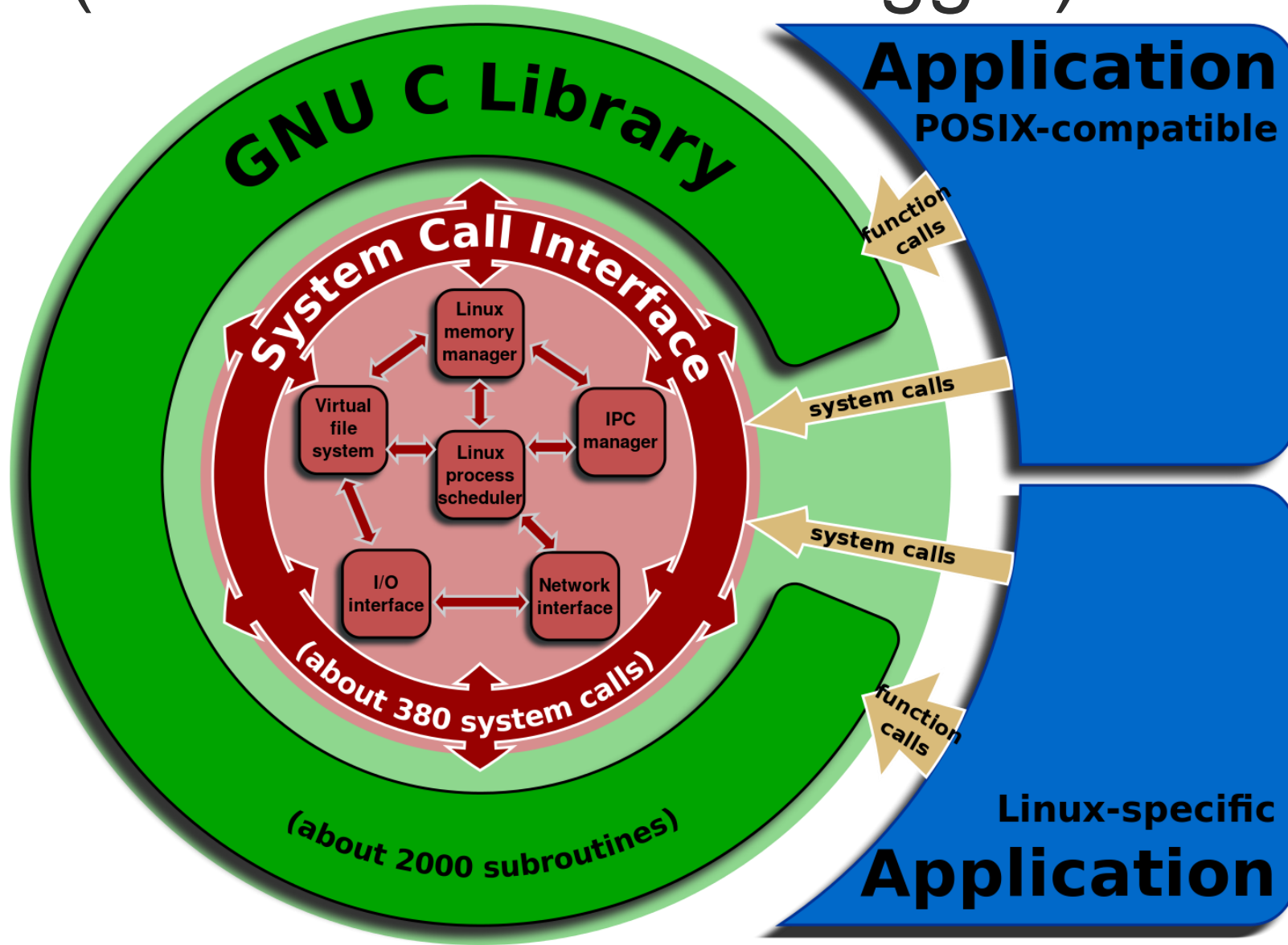
- This work represents the views of the author(s) and does not necessarily reflect the views of Microsoft or Google
- Linux is a registered trademark of Linus Torvalds.
- Other company, product, and service names may be trademarks or service marks of others.

# Outline

- What is POSIX?
- Why do these extensions matter?
- Demo
- What if we don't have them?
  - What works?
  - Some history: CIFS Extensions
  - Alternatives
- Some details
- What if Linux continues to extend, to improve?

# POSIX != Linux

(Linux API is much bigger)

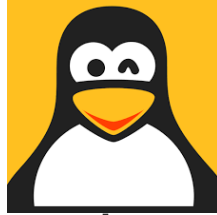


# Linux is BIG

- Currently 293 Linux syscalls!
- VS
- About 100 POSIX API calls



# 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

# What could you try today?

- For obvious reasons, these experimental changes are not turned on by default so ...
  - With current mainline Linux (4.18-rc)
  - You must mount with “vers=3.1.1”
  - AND also specify new mount option “posix” and turn off remapping of reserved characters (ie append “nomapposix”)
  - Only a few limited protocol features (posix open context request) can be tried but although small change it is VERY useful and enough to experiment with and test various apps
- JRA has a tree on samba.org ([git.samba.org/jra/samba/.git](https://git.samba.org/jra/samba/.git) in branch “master-smb2”) with prototype server code



# Example

- 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”

# Note the new mount option “posix” vs “nounix” (in default SMB3.1.1 mount)

```
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,noforceuid,gid=0,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,nounix,serverino,mapposix,rsize=1048576,wsz=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,wsz=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!

```
root@Ubuntu-17-Virtual-Machine: ~
File Edit View Search Terminal Help

root@Ubuntu-17-Virtual-Machine:~# ls /mnt-rename-test -la
total 2052
drwxr-xr-x  2 root root   0 May 31 18:19 .
drwxr-xr-x 34 root root 4096 May 31 18:13 ..
-rwxr-xr-x  1 root root   0 May 31 18:18 emptyfile
-rwxr-xr-x  1 root root   0 May 31 18:19 emptyfile-posix
-rwxr-xr-x  1 root root  16 May 31 18:17 targetfile
-rwxr-xr-x  1 root root  16 May 31 18:19 targetfile-posix
root@Ubuntu-17-Virtual-Machine:~# mount | grep rename
//localhost/rename-test on /mnt-rename-test type cifs (rw,relatime,vers=3.1.1,cache=strict,username=testuser,doma
root@Ubuntu-17-Virtual-Machine:~# mv /mnt-rename-test/emptyfile /mnt-rename-test/targetfile
mv: cannot move '/mnt-rename-test/emptyfile' to '/mnt-rename-test/targetfile': Permission denied

root@Ubuntu-17-Virtual-Machine:~# tail -f /mnt-rename-test/targetfile
targetfile data
tail: /mnt-rename-test/targetfile: No such file or directory
tail: no files remaining
root@Ubuntu-17-Virtual-Machine:~#
```

```
root@Ubuntu-17-Virtual-Machine: ~
File Edit View Search Terminal Help

root@Ubuntu-17-Virtual-Machine:~# mount | grep rename
//localhost/rename-test on /mnt-rename-test type cifs (rw,relatime,vers=3.1.1,cache=strict,username=testuser,doma
root@Ubuntu-17-Virtual-Machine:~# mv /mnt-rename-test/emptyfile-posix /mnt-rename-test/targetfile-posix
root@Ubuntu-17-Virtual-Machine:~#
```

## 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,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,seclimit=,uid=0,noforceuid,gid=0,noforcegid,addr=127.0.0.1,file_mode=0755,dir_mode=0755,soft,nosuid,nodev,posixpaths,serverino,mapposix,noperm,rsize=1048576,wsiz
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 (continued) – Neg response

Filter: smb2 Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
6	0.017898824	127.0.0.1	127.0.0.1	SMB2	354	Negotiate Protocol Response
8	0.018071383	127.0.0.1	127.0.0.1	SMB2	190	Session Setup Request, NTLMSSP
9	0.025255219	127.0.0.1	127.0.0.1	SMB2	360	Session Setup Response, Error:
10	0.025479327	127.0.0.1	127.0.0.1	SMB2	428	Session Setup Request, NTLMSSP
12	0.067445361	127.0.0.1	127.0.0.1	SMB2	142	Session Setup Response

Type: SMB2\_ENCRYPTION\_CAPABILITIES (0x0002)  
DataLength: 4  
Reserved: 00000000  
CipherCount: 1  
CipherId: AES-128-CCM (0x0001)  
▼ Negotiate Context: Unknown Type: (0x100)

Type: Unknown (0x0100)  
DataLength: 4  
Reserved: 00000000  
Unknown: 00000000

00c0	4a 00 d0 00 00 00 60 48	06 06 2b 06 01 05 05 02	J.....`H ..+.....
00d0	a0 3e 30 3c a0 0e 30 0c	06 0a 2b 06 01 04 01 82	.>0<..0. ..+.....
00e0	37 02 02 0a a3 2a 30 28	a0 26 1b 24 6e 6f 74 5f	7....*( .&.\$not
00f0	64 65 66 69 6e 65 64 5f	69 6e 5f 52 46 43 34 31	defined_ in RFC41
0100	37 38 40 70 6c 65 61 73	65 5f 69 67 6e 6f 72 65	78@pleas e ignore
0110	00 00 00 00 00 00 01 00	26 00 00 00 00 00 01 00	..... &.....
0120	20 00 01 00 02 f8 9d 50	88 6b ee f7 1d 8e 3b 78	.....P .k....;x
0130	9e 0e d6 91 ea 78 12 d2	15 ef e6 93 ca 67 52 2b	.....x.. .....gR+
0140	52 5f 4b 30 00 00 02 00	04 00 00 00 00 00 01 00	R_K0.....
0150	01 00 00 00 00 00 00 01	04 00 00 00 00 00 00 00	.....
0160	00 00		..

# Details continued – Create (POSIX) req

The image shows a Wireshark network traffic analysis window. The filter is set to 'smb2'. The packet list shows several SMB2 packets. The selected packet (No. 1) is a Create Request. The details pane shows the chain elements for this packet, including SMB2\_CREATE\_REQUEST\_LEASE and SMB2\_CREATE\_DURABLE\_HANDLE\_REQUEST. The hex dump at the bottom shows the raw data of the packet, with the ASCII column displaying the file path and other metadata.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	SMB2	382	Create Request File: newposixf
2	0.005646911	127.0.0.1	127.0.0.1	SMB2	390	Create Response File: newposixf
4	0.005807273	127.0.0.1	127.0.0.1	SMB2	174	GetInfo Request FILE_INFO/SMB2_
5	0.005988789	127.0.0.1	127.0.0.1	SMB2	150	GetInfo Response
6	0.006088712	127.0.0.1	127.0.0.1	SMB2	262	Create Request File: newposixf

► Chain Element: SMB2\_CREATE\_REQUEST\_LEASE "RqLS"

► Chain Element: SMB2\_CREATE\_DURABLE\_HANDLE\_REQUEST "DHnQ"

▼ Chain Element: <invalid> "5025ad93-b49c-e711-b423-83de968bcd7c"  
Chain Offset: 0x00000000

▼ Tag: 5025ad93-b49c-e711-b423-83de968bcd7c  
Offset: 0x00000010  
Length: 16

▼ Data  
Offset: 0x00000020  
Length: 4

Offset	Hex	ASCII
0d0	69 00 6c 00 65 00 00 00 00 00 00 00 00 00 50 00	i.l.e... ..P.
0e0	00 00 10 00 04 00 00 00 18 00 34 00 00 00 52 71	..... ..4...Rq
0f0	4c 73 00 00 00 00 32 91 57 d8 3b 99 47 1c a3 7d	Ls....2. W.;.G..}
100	a7 81 cc 01 a1 0e 07 00 00 00 00 00 00 00 00 00	.....
110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
120	00 00 00 00 00 00 00 00 00 00 00 00 00 28 00	..... (.
130	00 00 10 00 04 00 00 00 18 00 10 00 00 00 44 48	..... ..DH
140	6e 51 00 00 00 00 00 00 00 00 00 00 00 00 00 00	nQ.....
150	00 00 00 00 00 00 00 00 00 00 10 00 10 00 00 00	.....
160	20 00 04 00 00 00 93 ad 25 50 9c b4 11 e7 b4 23	..... %P.....#
170	83 de 96 8b cd 7c a4 81 00 00 00 00 00 00 00 00	..... .....

# Details continued – create response

The image shows a Wireshark network traffic analysis interface. The top toolbar includes icons for menu, settings, capture, analyze, statistics, telephony, tools, network, help, and a search icon. Below the toolbar, the filter is set to 'smb2'. The main display area shows a list of network packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. Packet 2 is highlighted in orange and is a 'Create Response' for file 'newposixfi'. Below the packet list, the details pane shows the structure of the response, including a 'Chain Element' (invalid), a 'Tag' (5025ad93-b49c-e711-b423-83de968bcd7c), and a 'Data' field. The bottom pane shows the raw packet data in hexadecimal and ASCII. The status bar at the bottom indicates 'Tag of chain entry (smb2 tag)', 'Packets: 18 · Displayed: 12 (66)', and 'Profile: Default'.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	SMB2	382	Create Request File: newposixfi
2	0.005646911	127.0.0.1	127.0.0.1	SMB2	390	Create Response File: newposixfi
4	0.005807273	127.0.0.1	127.0.0.1	SMB2	174	GetInfo Request FILE_INFO/SMB2_
5	0.005988789	127.0.0.1	127.0.0.1	SMB2	150	GetInfo Response
6	0.006088712	127.0.0.1	127.0.0.1	SMB2	262	Create Request File: newposixfi

Length: 52  
  ▶ LEASE\_V2  
  ▼ Chain Element: <invalid> "5025ad93-b49c-e711-b423-83de968bcd7c"  
    Chain Offset: 0x00000000  
    ▼ Tag: 5025ad93-b49c-e711-b423-83de968bcd7c  
      Offset: 0x00000010  
      Length: 16  
      ▼ Data  
        Offset: 0x00000020  
        Length: 56

```
00e0 00 00 10 00 04 00 00 00 18 00 34 00 00 00 52 71 ..... .4...Rq
00f0 4c 73 00 00 00 00 32 91 57 d8 3b 99 47 1c a3 7d Ls....2. W.;.G..}
0100 a7 81 cc 01 a1 0e 07 00 00 00 00 00 00 00 00 .....
0110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0120 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 .....
0130 00 00 10 00 10 00 00 00 20 00 38 00 00 00 93 ad ..... .8.....
0140 25 50 9c b4 11 e7 b4 23 83 de 96 8b cd 7c 01 00 %P.....# .....|..
0150 00 00 00 00 00 00 a4 01 00 00 01 05 00 00 00 00 .....
0160 00 05 15 00 00 00 d1 ce 97 47 26 98 0e 65 f1 e2 ..... .G&..e..
0170 40 93 e8 03 00 00 01 02 00 00 00 00 00 16 02 00 @.....
0180 00 00 e8 03 00 00 .....
```

Tag of chain entry (smb2 tag)    Packets: 18 · Displayed: 12 (66)    Profile: Default

# Summary of What works

- Without Extensions
- With Extensions

Other Alternatives: AAPL

# Note that Apple create context (AAPL) can be used for some of this

The image shows a Wireshark capture of SMB2 traffic. The top part is a packet list table, and the bottom part is a packet details pane for packet 254.

No.	Time	Source	Destination	Protocol	Info
246	9.468750	10.10.10.116	10.10.10.30	SMB2	Create Request File: ;Close Request
248	9.471618	10.10.10.30	10.10.10.116	SMB2	Create Response File: [unknown];Close Re
250	9.472478	10.10.10.116	10.10.10.30	SMB2	Create Request File: file;GetInfo Reques
252	9.476572	10.10.10.30	10.10.10.116	SMB2	Create Response File: file;GetInfo Respc
254	9.476759	10.10.10.116	10.10.10.30	SMB2	Create Request File: file:com.apple.Laur

Disposition: Open (if file exists open it, else fail) (1)

- ▶ Create Options: 0x00000001
- ▼ Filename:
  - Offset: 0x00000078
  - Length: 0
- ▼ ExtraInfo SMB2\_AAPL\_CREATE\_CONTEXT
  - Offset: 0x00000080
  - Length: 40
    - ▼ Chain Element: SMB2\_AAPL\_CREATE\_CONTEXT "AAPL"
      - Chain Offset: 0x00000000
        - ▼ Tag: AAPL
          - Offset: 0x00000010
          - Length: 4
        - ▼ Data: AAPL Create Context request
          - Offset: 0x00000018
          - Length: 16
            - ▼ AAPL Create Context request
              - Command code: Resolve ID (2)
              - Reserved: 0x00000000
              - File Id: 0x0000000000760a9c

- ▼ SMB2 (Server Message Block Protocol version 2)
- ▼ SMB2 Header
  - Server Component: SMB2
  - Header Length: 64
  - Credit Charge: 1

0040 03 90 00 00 01 00 fe 53 4d 42 40 00 01 00 00 00 .....S MB@.....  
0050 00 00 05 00 00 01 00 00 00 00 a8 00 00 00 75 00 .....U.  
0060 00 00 00 00 00 00 ff fe 00 00 02 00 00 00 06 00 .....  
0070 00 00 81 09 4a 70 00 00 00 00 00 00 00 00 00 .....Jp..  
0080 00 00 00 00 00 00 39 00 00 00 02 00 00 00 00 00 .....9.  
0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80 00 .....  
00a0 10 00 10 00 00 00 07 00 00 00 01 00 00 00 01 00 .....  
00b0 00 00 78 00 00 00 80 00 00 00 28 00 00 00 00 00 ...x....(.....

# And the response:

smb2

No.	Time	Source	Destination	Protocol	Info
246	9.468750	10.10.10.116	10.10.10.30	SMB2	Create Request File: ;Close Request
248	9.471618	10.10.10.30	10.10.10.116	SMB2	Create Response File: [unknown];Close
250	9.472478	10.10.10.116	10.10.10.30	SMB2	Create Request File: file;GetInfo Requ
252	9.475572	10.10.10.30	10.10.10.116	SMB2	Create Response File: file;GetInfo Res

Create Action: The file existed and was opened (1)  
Create: Apr 2, 2014 09:46:43.000000000 CDT  
Last Access: Mar 2, 2016 11:16:36.000000000 CST  
Last Write: Apr 28, 2016 10:35:07.000000000 CDT  
Last Change: Apr 28, 2016 10:35:07.000000000 CDT  
Allocation Size: 0  
End Of File: 0

- ▶ File Attributes: 0x00000010
- ▶ GUID handle File: [unknown]

▼ ExtraInfo SMB2\_AAPL\_CREATE\_CONTEXT

- Offset: 0x00000098
- Length: 48
- ▼ Chain Element: SMB2\_AAPL\_CREATE\_CONTEXT "AAPL"
  - Chain Offset: 0x00000000
  - ▼ Tag: AAPL
    - Offset: 0x00000010
    - Length: 4
  - ▼ Data: AAPL Create Context response
    - Offset: 0x00000018
    - Length: 24
    - ▼ AAPL Create Context response
      - Command code: Resolve ID (2)
      - Reserved: 0x00000000
      - NT Status: STATUS\_SUCCESS (0x00000000)
      - Server path: file

▼ SMB2 (Server Message Block Protocol version 2)

SMB2 Header

00d0	00 00 00 00 00 00 98 00	00 00 30 00 00 00 00 00	.....0.....
00e0	00 00 10 00 04 00 00 00	18 00 18 00 00 00 41 41	.....AA
00f0	50 4c 00 00 00 00 02 00	00 00 00 00 00 00 00 00	PL.....
0100	00 00 08 00 00 00 66 00	69 00 6c 00 65 00 fe 53	.....f.ile.S
0110	4d 42 40 00 01 00 00 00	00 00 06 00 00 01 05 00	MB@.....
0120	00 00 00 00 00 00 76 00	00 00 00 00 00 00 ff fe	.....v.....

# 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?
  - What does it really need from SMB3 to be optimal...?
  - Not just to do 'cool' things: compile kernel on SMB3 mount, boot linux (show blazing performance ...!)
  - For all key features: SMB3  $\geq$  CIFS with/Unix Extensions
    - We are not asking user to go backwards
  - Can we extend them as Linux API moves
    - (Did we mention that mount API and fsinfo/statfs BOTH are changing – see Al Viro's git tree ... and that statx was added last year and Linux continues to evolve ...)

# 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 JRA's Server Perspective?

- Learn from the mistakes of SMB1 Unix extensions.
  - Security issues paramount.
  - Remove the possibility of server-followed symlinks
    - Break interoperability with NFS :-), but necessary.
- Minimum Necessary Change (with apologies to Asimov's "*The End of Eternity*").
  - Fewer changes to the protocol the better.
  - Use the fact that we have experience with Samba in sharing between Windows and UNIX SMB connections.

# Server Perspective Continued..

- Server-followed symlinks that the client can create have been a security disaster in Samba.
- Server-following symlinks is a useful holdover from ancient times, when admin-created symlinks gave great flexibility to setups.
  - As soon as clients gained the ability via UNIX extensions to create symlinks, disaster strikes.
  - Failed design decision to store these as real symlinks on the server filesystem.
    - Convenience for dual NFS / SMB1 servers.
- **THIS MUST NOT BE ALLOWED FOR SMB2+**

# Server Perspective Continued..

- The key for SMB2 UNIX extensions is to allow simultaneous Windows and UNIX handles – using SMB2 create contexts.
  - Adding UNIX extension create context turns on POSIX behavior for this handle only.
  - Allows client code to probe for POSIX behavior – SMB2 specifies unknown create contexts are ignored.
  - The Samba server already has to handle this case in serving POSIX and non-POSIX client simultaneously.
- Leads to new Negotiate context requirement from the server.
  - That way a client can determine if a server could support POSIX behavior on a handle, but chooses not to.
  - POSIX servers may expose POSIX behaviors or deny them depending on pathname (crossing mount points).



# Server Perspective Continued..

- The rest of the changes are relatively small.
- One new info level needed to cope with POSIX stat returns.
- Keep protocol as close to “native” Windows as possible.
  - Map POSIX ‘mode’ into Windows ACL encoding.
  - No POSIX ACLs – return everything as Windows ACLs.
  - No POSIX uid/gids – return everything as Windows SIDs.
    - Client systems must cope with mapping SIDs anyway.
- Filename handling (POSIX specific, case sensitive) is the largest change. No access to Windows streams.
  - If you want a Windows stream handle, open a Windows stream handle.
  - Keep USC2 encoding (no change from Windows). UTF-8 would be nice, but not strictly required so drop it.
- Allow server to associate modified behavior on a per-handle basis.

Details

C U V

# Proposed SMB3 POSIX Extensions

- Negotiate Protocol
  - SMB3.1.1 (or later required)
    - POSIX Negotiate Context 0x100
    - Version is implied by the context (in case extensions are revised in the future to a version 2 or 3 ...) but there is a reserved field that can be used in emergency
  - 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>)
  - **NEED TO VERIFY:** 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

# Other

- Hardlinks use the Windows setinfo call (already used by cifs.ko etc)
- Symlinks are client-only (opaque to server) can use “mfsymlinks” (as Mac and cifs.ko already do) or the Windows NFS symlink reparse point. Servers do not follow these symlinks (for obvious security reasons)
- Other linux extensions, e.g. fallocate are mapped to existing SMB3 operations where possible

# Proposed 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

# POSIX open/create context resp.

- `__u32 number_of_hardlinks`
- `__u32 type; /* e.g. REPARSE tag */`
- `__u32 perms; /* mode & ~S_IFMT */`
- `struct dom_sid sid_owner;`
- `struct dom_sid sid_group;`

# SMB2/SMB3 Create Contexts

We define a new context name for this new CreateContext to distinguish it from others like MxAc and RqLs and a buffer to include POSIX

Information in request and response

**SMB2\_CREATE\_TAG\_POSIX =**

**"\x93\xAD\x25\x50\x9C\xB4\x11\xE7\xB4\x23\x83\xDE\x96\x8B\xCD\x7C"**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Next																															
NameOffset																NameLength															
Reserved																DataOffset															
DataLength																															
Buffer (variable)																															
...																															



# Proposed POSIX Infolevels

- Query/SetInfo and Query\_DIR
  - 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

# Also need to support statfs (“stat -f”) currently using FS\_INFO level 100

```
+struct posix_v1_query_fs_info_response {
    /* Returned for context SMB2_POSIX_V1_STATFS_INFO */
    /* EXISTING posix extensions for fs info is good enough, note For undefined recommended transfer size return -1
in that field */
    __le32 OptimalTransferSize; /* bsize on some os, iosize on other os */
    __le32 BlockSize; /* f_frsize, disk bytes avail based on this size */
    /* Next three fields are in terms of the block size above. If block size unknown, 4096 would be reasonable block
size for a server to report. Note that returning blocks/blocksavail removes need to make second call (to QFSInfo
level 0x103. UserBlockAvail is typically less than or equal to BlocksAvail, if no distinction is made return the same
value in each */
    __le64 TotalBlocks;
    __le64 BlocksAvail; /* bfree */
    __le64 UserBlocksAvail; /* bavail */
    __le64 TotalFileNodes;
    __le64 FreeFileNodes;
    __le64 FileSysIdentifier; /* fsid */
    /* NB Namelen comes from FILE_SYSTEM_ATTRIBUTE_INFO call , and flags can come from
FILE_SYSTEM_DEVICE_INFO call */
    /* In Linux f_type is always 0xFE 'S' 'M' 'B' since that is the fs, not the server's os – so server does not have to
return it */
```

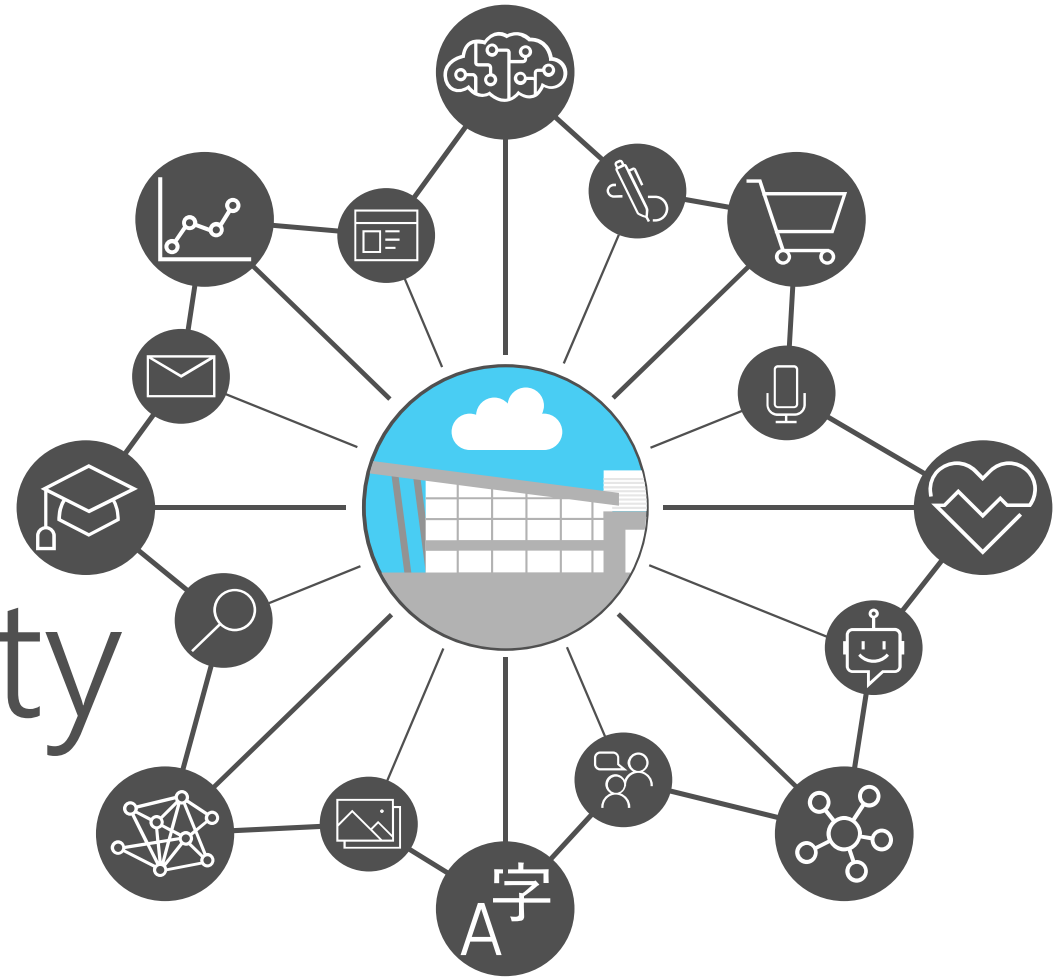
# Wireshark

- 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 on enhancing readdir this week
- Continue extending the wireshark dissectors (see Aurelien)
- Continue testing/prototyping here and additional testing at SNIA in the fall
- Continue updating the wiki with details:  
<https://wiki.samba.org/index.php/SMB3-Linux>

# Redmond Interoperability Plugfest 2018



# Thank you