Linux CIFS client year in review: From Nocturnal Monster Puppies to Funky Weasels

Steve French
CIFS maintainer
Samba team
Senior Engineer
IBM Linux Technology Center

Legal Statement

This work represents the views of the author and does not necessarily reflect the views of IBM Corporation
A full list of U.S. trademarks owned by IBM may be found at http://www.ibm.com/legal/copytrade.shtml.
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 or sfrench@us.ibm.com)
  - Author and maintainer of Linux cifs vfs (for accessing Samba, Windows and various SMB/CIFS based NAS appliances)
  - Member of the Samba team, coauthor of CIFS Technical Reference and former SNIA CIFS Working Group chair
  - Architect: Filesystems/NFS/Samba IBM LTC

Outline

- Why SMB/CIFS ... 24 years and counting?
- Highlights
  - Kerberos
  - DFS
  - Ipv6
  - ACLs
- Unix Extensions ... good enough?
  - Why were they developed?
  - What and where are they?
- Something missing ...
  - What about SMB2?
  - What about more Extensions ...?

CIFS Rocks On...

- Windows goes on and on - sees new Vistas
- Other servers from many companies
  - Samba 3.0.28a, 3.2, 4 (Novell, RedHat, IBM SOFS and Nitix ...)
  - NetApp ...
- And many clients
  - Smbclient, HPUX
  - Linux CIFS VFS
  - JCIFS, MacOS ...

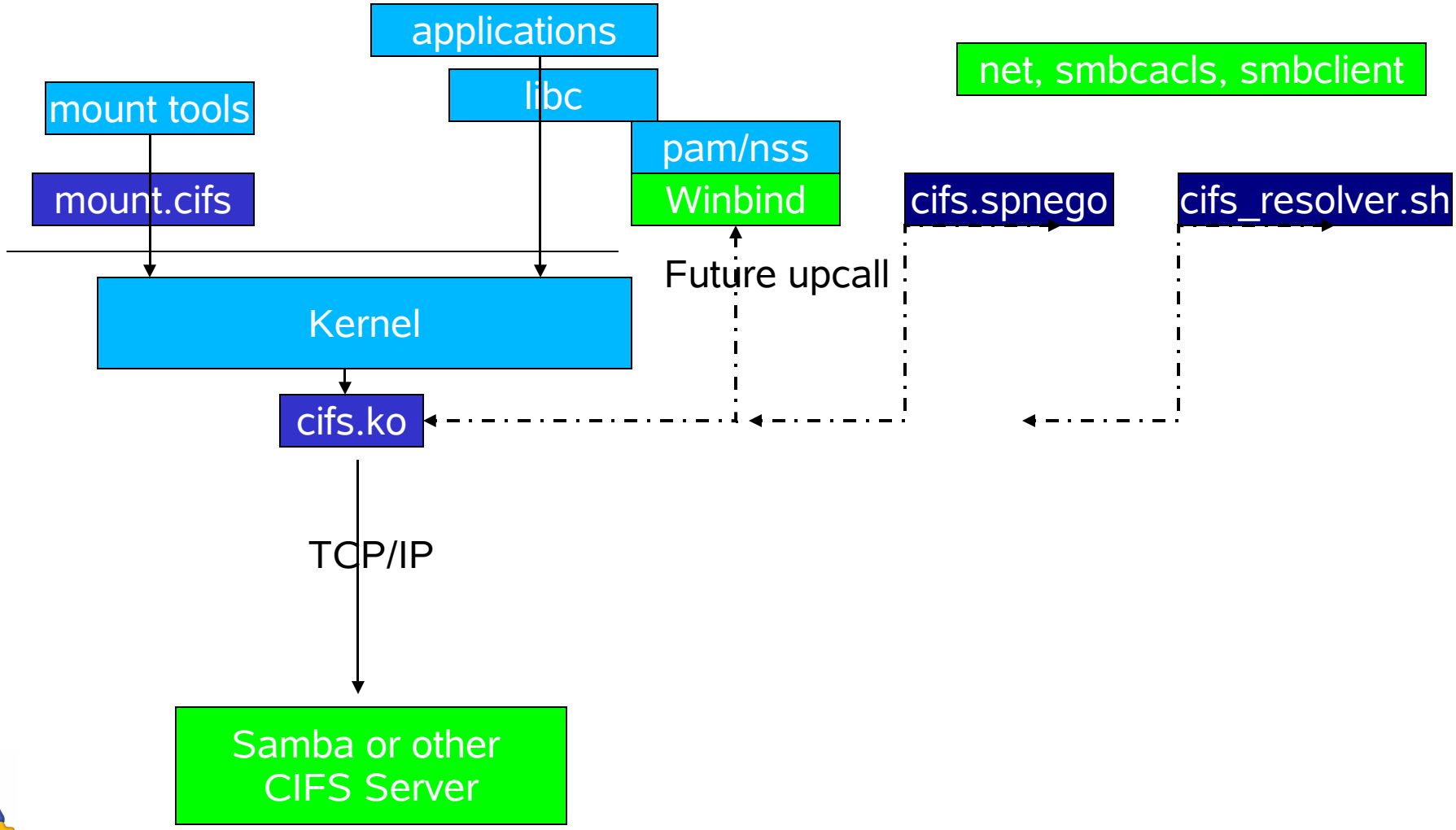
Goals ...

- Full local/remote transparency desired
- Need near perfect POSIX semantics over cifs
- Be fast, efficient, full function gateway to accessing data on Windows and Samba servers
- Other ongoing requirements
  - Better caching of directory information
  - Improved DFS (distributed name space)
  - Better large file sequential performance
  - Better recovery after network failure
  - QoS

And the alternatives?

- NFS v3 or v4
- AFS/DFS
- HTTP/WebDav
- Cluster Filesystem Protocols

CIFS and related components



Last year at this time: Status

- Linux CIFS client
  - Version 1.48 (Linux 2.6.21 Nocturnal Monster Puppy) Two years ago at this time ... cifs version 1.42
  - (1.43 included the much improved POSIX locking)
  - Version 1.32 included POSIX ACLs, statfs, lsattr
- Smbclient
  - Samba 3.0.25 includes client test code for POSIX locking, POSIX open/unlink/mkdir.
- HP/UX client also supports Unix Extensions
- Sun is developing a kernel CIFS client for Solaris
- Server
  - Samba 3.0.25 includes POSIX Locking (POSIX ACLs, QFSInfo, Unix Extensions implemented before) and POSIX open/unlink/mkdir.

Now ... Status

- Linux CIFS client
  - Version 1.52, Linux 2.6.25-rc9(!) Funky Weasel is Jiggy wit it (?!)
  - A year ago at this time...cifs version 1.48 and kernel version 2.6.21
- Smbclient
  - Samba 3.0.28a includes dfs support, per tcon encryption
- Sun kernel CIFS server for Solaris in development
- Huge amounts of Microsoft documentation promise more for the future only obstacle is time for perfect interoperability ... (contributions welcome)
- Server: year of the ctdb ...
  - Samba 3.0.28a, more Unix Extensions implemented including per tcon encryption
  - ctdb and Samba 3.2 much improved clustering support and performance (receivefile and more)

Last year at this time: A year in review for the client

- 2006-2007 Growing fast (well over 100 changesets per year ...), one of the larger (22KLOC) kernel filesystems
- Write performance spectacularly better on 3 of 11 iozone cases
- POSIX locking, lock cancellation support (and much better POSIX byte range lock emulation to Windows)
- NTLMv2 (much more secure authentication, and new “sec=” mount options)
- Older server support (OS/2, Windows 9x)
- “deep tree” mounts
- New mkdir reduces 50% of network requests for this op
- Improved atime/mtime handling (and better performance)
- Improved POSIX semantics (lots of small fixes)
- Can be used for home directory now ... everything should work!

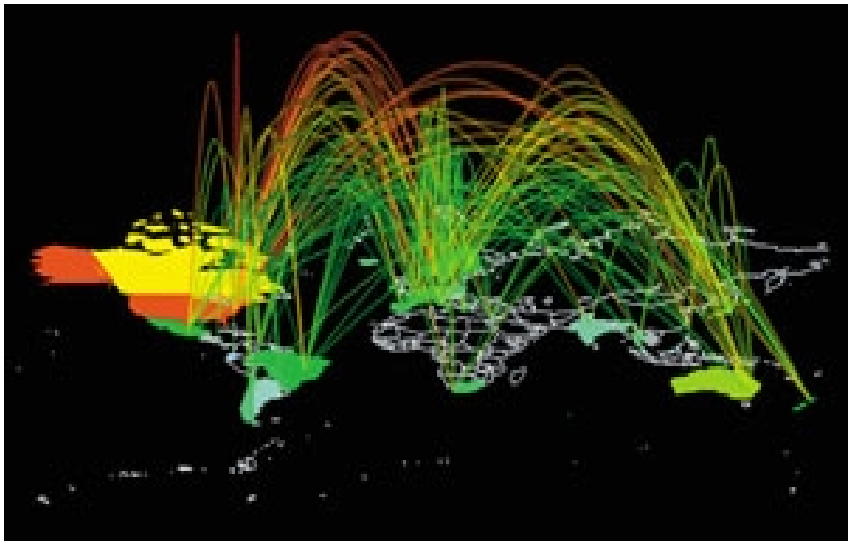
A year in review for the client

- 2007-2008 Growing faster (195 changesets from 44 developers)
- One of larger Linux kernel file systems (24KLOC up about 10%, and over 1/3, more than 8K added, rewritten, cleaned up, “git log -p” output (patches) is over 1.4MB
- Experimental Kerberos support added
- Experimental DFS support added
- cifsacl support (query mode and chmod use ACL ops)
- Ipv6 support (code started at last SambaXP)
- Improved POSIX semantics (lots of small fixes): allow uid/gid override even for Unix servers, add new “nounix” mount option
- Add posix unlink (still working on posix open changes)
- support for pipe open over IPC$
- nfsd over cifs supported in some cases
- Very large read (127K) support to Samba

Kerberos support

- Developed with assistance of RedHat and others
- Requires additional user space helper util (in Samba 3 source tree)
- Experimental – probably will remove experimental flag by 2.6.27

DFS (Global Namespace) improvements

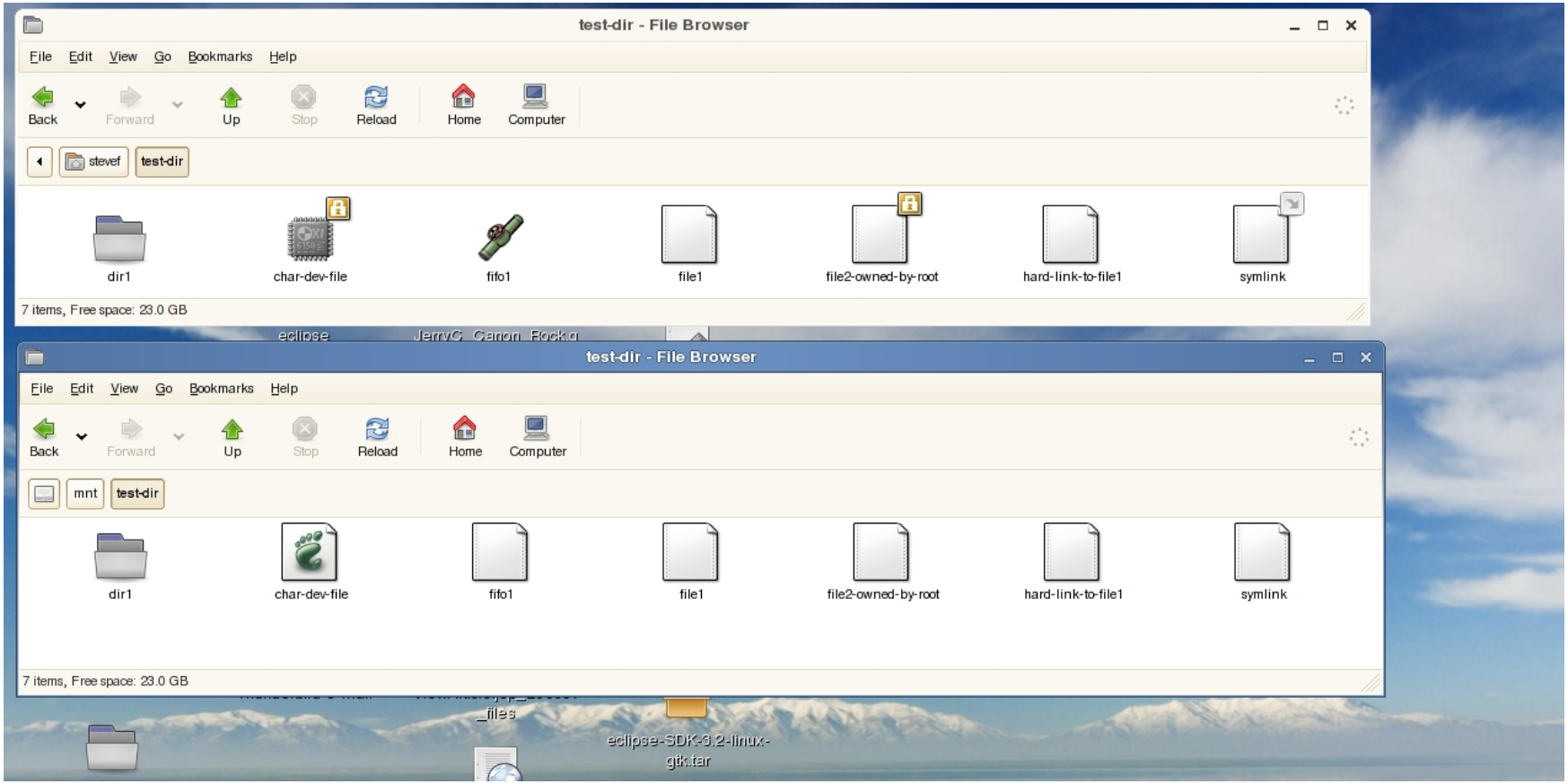


- DFS patch integrated, needs some cleanup
- We need to improve ability to find nearest replica, and recover after failure
- And also to hint "least busy" server for load balancing

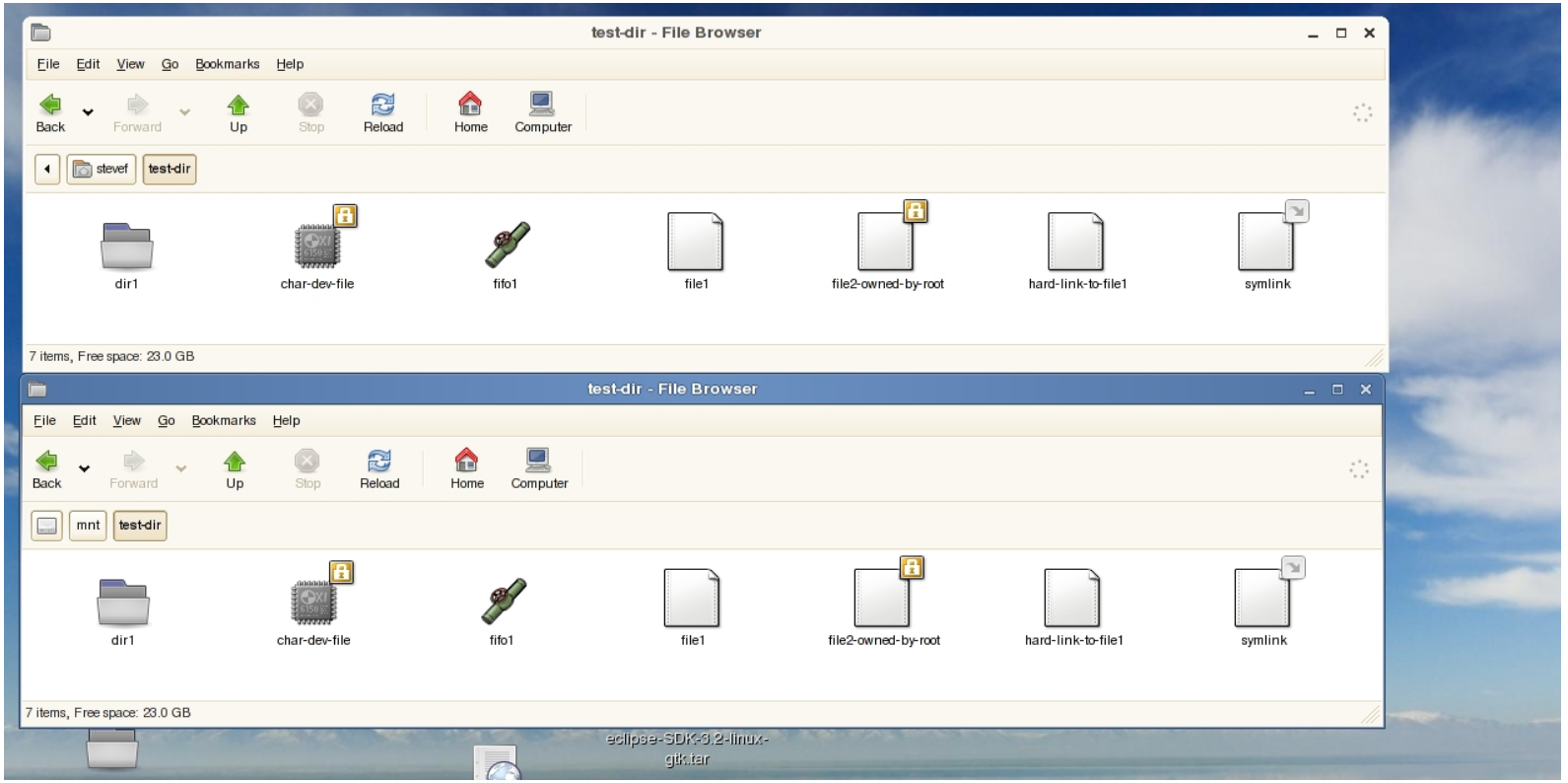
Quick review: CIFS Unix Extensions

- Developed/Documented by HP (extending early work by SCO) and others then documented by SNIA in the CIFS Technical Reference
  - Required only modest extensions to server
  - Solved key problems for POSIX clients including:
    - How to return: UID/GID, mode
    - How to handle symlinks
    - How to handle special files (devices/fifos)

Without CIFS extensions, less local/remote transparency...



Much improved with CIFS Extensions



What about SFU approach?

- Lessons from SFU:
  - Map mode, group and user (SID) owner fields to ACLs
  - Do hardlinks via NT Rename
  - Get inode numbers
  - Remap illegal characters to Unicode reserved range
  - FIFOs and device files via OS/2 EAs on system files
- OK, **but not good enough** &
  - Some POSIX byte range lock tests fail
  - Semantics are awkward for symlinks, devices
  - UID mapping a mess
  - Performance slow
  - Operations much less atomic and not robust enough
  - Rename/delete semantics are hard to make reliable

CIFS Unix Extensions

- Problem ... a lot was missing:
  - Way to negotiate per mount capabilities
  - POSIX byte range locking
  - ACL alternative (such as POSIX ACLs)
  - A way to handle some key fields in statfs
  - Way to handle various newer vfs entry points
    - lsattr/chattr
    - Inotify
    - New xattr (EA) namespaces

Original Unix Extensions Missing POSIX ACLs and statfs info

```
smf-t41p:/home/stevef # getfacl /mnt/test-dir/file1
# file: mnt/test-dir/file1
# owner: root
# group: root
user::rwx
group::rw-
other::rwx

smf-t41p:/home/stevef # stat -f /mnt1
  File: "/mnt1"
    ID: 0        Namelen: 4096    Type: UNKNOWN
(0xff534d42)
Block size: 1024       Fundamental block size: 1024
Blocks: Total: 521748     Free: 421028     Available:
421028
Inodes: Total: 0          Free: 0
```

With CIFS POSIX Extensions, ACLs and statfs better

```
smf-t41p:/home/stevef # getfacl /mnt/test-dir/file1
# file: mnt/test-dir/file1
# owner: stevef
# group: users
user::rw-
user:stevef:r--
group::r--
mask::r--
other::r--

smf-t41p:/home/stevef # stat -f /mnt1
  File: "/mnt1"
    ID: 0        Namelen: 4096    Type: UNKNOWN (0xff534d42)
Block size: 4096       Fundamental block size: 4096
Blocks: Total: 130437     Free: 111883     Available: 105257
Inodes: Total: 66400      Free: 66299
```

POSIX Locking

- Locking semantics differ between CIFS and POSIX at the application layer.
  - CIFS locking is mandatory, POSIX advisory.
  - CIFS locking stacks and is offset/length specific, POSIX locking merges and splits and the offset/lengths don't have to match.
  - CIFS locking is unsigned and absolute, POSIX locking is signed and relative.
  - POSIX close destroys all locks.

Last year … new features in srv

- POSIX OPEN/CREATE/MKDIR
- POSIX “who am I” (on this connection)
- POSIX stat/lookup
- Under development (3.0.27+ ?) -
  - CIFS transport encryption (GSSAPI encrypt at the CIFS packet level).
  - Based on authenticated user (vuid) – encryption context per user.
  - Allows mandatory encryption per share.

How did we do on Roadmap from last year?

- Client
  - 2.6.22 included new mkdir/open (Y)
- Server
  - Samba 3.0.25 was completed. (Y)
    - Encryption feature developed. (Y, but Server only)
  - Samba 4 Unix/POSIX Extensions started with new POSIX CIFS client backend
- In discussions with other client and server vendors about feature needs (Y, continuing. Good progress at SNIA and Google conferences)

Do we still need more new POSIX extensions: e.g. POSIX Errors

- NT Status codes (16 bit error nums) already has a reserved range
  - 0xF3000000 + POSIX errnum
  - POSIX errnum vary in theory, but not much in practice for common ones use
  - POSIX errnums fixed
  - New capability(will probably be)
    - #define CIFS_UNIX_POSIX_ERRORS 0x20
  - Do we need to define new errmapping SMB for client to resolve unknown POSIX errors backs to NT Status?

Beating the competition - NFSv4

- Key differences
  - CIFS is richer protocol (huge variety of network filesystem functions available in popular servers)
  - CIFS supports Windows and POSIX model through different commands as necessary
  - Detailed CIFS documentation available (no more secrets ...?)
  - CIFS can negotiate features with more flexibility: on a “tid” not just a session (or RPC pipe). This is helpful in tiered/gateway/clustered environments
  - CIFS does not have SunRPC baggage
  - And we have the Samba team ...
- And we are easier to configure than most cluster filesystems ...

Near term priorities on client side

- Digesting large amounts of Microsoft documentation, looking for any problems, bugs
- Finish up of DFS patch
- More kerberos testing
- Finish up of POSIX Open (big performance boost in some operations)
- Improved large write support (increase iovec – so more than 56K writes)
- Finish up of pipe opens over IPC$ (help WINE and others who want named pipe support)
- Additional performance analysis
- ... and your requests!

Where to go from here?

- Discussions on samba-technical and linux-cifs-client mailing lists
- Wire layout is visible in fs/cifs/cifspdu.h
- Working on updated draft reference document for these cifs protocol extensions
- See http://samba.org/samba/CIFS_POSIX_extensions.html

Thank you for your time!