v9fs: Plan 9 Resource Sharing for Linux

ABOUT

v9fs is a Unix implementation of the Plan 9 9p remote filesystem protocol.

This software was originally developed by Ron Minnich <rminnich@sandia.gov> and Maya Gokhale. Additional development by Greg Watson <gwatson@lanl.gov> and most recently Eric Van Hensbergen <ericvh@gmail.com>, Latchesar Ionkov <lucho@ionkov.net> and Russ Cox <rsc@swtch.com>.

The best detailed explanation of the Linux implementation and applications of the 9p client is available in the form of a USENIX paper:
http://www.usenix.org/events/usenix05/tech/freenix/hensbergen.html

Other applications are described in the following papers:
* XCPU & Clustering
  http://xcpu.org/papers/xcpu-talk.pdf
* KVMFS: control file system for KVM
  http://xcpu.org/papers/kvmfs.pdf
* CellFS: A New Programming Model for the Cell BE
  http://xcpu.org/papers/cellfs-talk.pdf
* PROSE I/O: Using 9p to enable Application Partitions
  http://plan9.escet.urjc.es/iwp9/cready/PROSE_iwp9_2006.pdf

USAGE

For remote file server:

```
mount -t 9p 10.10.1.2 /mnt/9
```

For Plan 9 From User Space applications (http://swtch.com/plan9)

```
mount -t 9p `namespace`/acme /mnt/9 -o trans=unix,uname=$USER
```

For server running on QEMU host with virtio transport:

```
mount -t 9p -o trans=virtio <mount_tag> /mnt/9
```

where mount_tag is the tag associated by the server to each of the exported mount points. Each 9P export is seen by the client as a virtio device with an associated "mount_tag" property. Available mount tags can be seen by reading /sys/bus/virtio/drivers/9pnet_virtio/virtio<n>/mount_tag files.

OPTIONS

trans=name select an alternative transport. Valid options are currently:
- unix - specifying a named pipe mount point
- tcp - specifying a normal TCP/IP connection
- fd - used passed file descriptors for connection

(see rfdno and wfdno)
virtio - connect to the next virtio channel available
(from QEMU with trans_virtio module)
rdma - connect to a specified RDMA channel

uname=name user name to attempt mount as on the remote server. The server may override or ignore this value. Certain user names may require authentication.

aname=name aname specifies the file tree to access when the server is offering several exported file systems.

cache=mode specifies a caching policy. By default, no caches are used.
loose = no attempts are made at consistency, intended for exclusive, read-only mounts
fscache = use FS-Cache for a persistent, read-only cache backend.

debug=n specifies debug level. The debug level is a bitmask.
0x01 = display verbose error messages
0x02 = developer debug (DEBUG_CURRENT)
0x04 = display 9p trace
0x08 = display VFS trace
0x10 = display Marshalling debug
0x20 = display RPC debug
0x40 = display transport debug
0x80 = display allocation debug
0x100 = display protocol message debug
0x200 = display Fid debug
0x400 = display packet debug
0x800 = display fscache tracing debug

rfdno=n the file descriptor for reading with trans=fd

wfdno=n the file descriptor for writing with trans=fd

maxdata=n the number of bytes to use for 9p packet payload (msize)

port=n port to connect to on the remote server

noextend force legacy mode (no 9p2000.u or 9p2000.L semantics)

version=name Select 9P protocol version. Valid options are:
9p2000 - Legacy mode (same as noextend)
9p2000.u - Use 9P2000.u protocol
9p2000.L - Use 9P2000.L protocol

dfltuid attempt to mount as a particular uid

dfltgid attempt to mount with a particular gid

afid security channel - used by Plan 9 authentication protocols

nodevmap do not map special files - represent them as normal files. This can be used to share devices/named pipes/sockets between hosts. This functionality will be expanded in later versions.

access	there are three access modes. user = if a user tries to access a file on v9fs filesystem for the first time, v9fs sends an attach command (Tattach) for that user. This is the default mode. <uid> = allows only user with uid=<uid> to access the files on the mounted filesystem any = v9fs does single attach and performs all operations as one user
cachetag	cache tag to use the specified persistent cache. cache tags for existing cache sessions can be listed at /sys/fs/9p/caches. (applies only to cache=fscache)

RESOURCES
=========

Our current recommendation is to use Inferno (http://www.vitanuova.com/inferno) as the 9p server. You can start a 9p server under Inferno by issuing the following command:

```
; styxlisten -A tcp!*!564 export '#U*'
```

The -A specifies an unauthenticated export. The 564 is the port # (you may have to choose a higher port number if running as a normal user). The '#U*' specifies exporting the root of the Linux name space. You may specify a subset of the namespace by extending the path: '#U*'/tmp would just export /tmp. For more information, see the Inferno manual pages covering styxlisten and export.

A Linux version of the 9p server is now maintained under the npfs project on sourceforge (http://sourceforge.net/projects/npfs). The currently maintained version is the single-threaded version of the server (named spfs) available from the same SVN repository.

There are user and developer mailing lists available through the v9fs project on sourceforge (http://sourceforge.net/projects/v9fs).

A stand-alone version of the module (which should build for any 2.6 kernel) is available via (http://github.com/ericvh/9p-sac/tree/master)

News and other information is maintained on SWiK (http://swik.net/v9fs) and the Wiki (http://sf.net/apps/mediawiki/v9fs/index.php).

Bug reports may be issued through the kernel.org bugzilla (http://bugzilla.kernel.org)

For more information on the Plan 9 Operating System check out http://plan9.bell-labs.com/plan9

For information on Plan 9 from User Space (Plan 9 applications and libraries ported to Linux/BSD/OSX/etc) check out http://swtch.com/plan9

STATUS
======

The 2.6 kernel support is working on PPC and x86.

PLEASE USE THE KERNEL BUGZILLA TO REPORT PROBLEMS. (http://bugzilla.kernel.org)