Past, present and future of Tcl virtual filesystems

Steve Huntley

Tcl Conference 2006

Perspective:

Virtual filesystems are the killer app for Tcl total world domination.

- Virtual filesystems are the ultimate glue, thus a natural fit for Tcl.
- In a world where every OS provides a hierarchical filesystem, and every program uses it, virtual filesystems are a universally applicable abstraction.

The past:

File transport:

- FTP, HTTP
- Every file transport protocol can look like every other.
- Even Windows Explorer does it.

File archiving:

- TAR, ZIP.
- Metakit vfs for starkits.

Drawbacks of the past:

- Emphasis on read functions

- Incomplete implementations

Challenges of the present:

- A VFS should be reflexive (one VFS should be able to serve as info source for another VFS)
- A VFS should be graftable as a branch of another VFS
- A VFS should have scalable performance
- i.e., a VFS should be as complete a metaphor as possible

A template virtual filesystem:

- Has no function except to be a complete metaphor:
  - read/write
  - all file info services supported
  - complete error handling
  - adequate performance

- A baseline for future development

New virtual filesystems developed on the template:

- a collating VFS (multiple locations appear as one)
- an SSH VFS (turns remote SSH server into file server)
- a quota VFS (imposes quotas on file attributes)
- a chroot VFS (makes all but a specified subdirectory invisible to interpreter)
- a versioning VFS (preserves all file edits committed)
- a delta VFS (designed to work with versioning VFS, generates deltas of file edits to save disk space)

- Virtual filesystems based on the template can be aggressively combined, stacked, daisy-chained, grafted.
- Data can be transformed in multiple discrete steps.
- Analog to Unix pipes and streams, but maps to hierarchical data!

The future:

Hierarchical data management:

Virtual filesytems can be used to manage not just files, but as a simplifying metaphor for handling any tree-structure data:

- LDAP
- XML
- Software objects

New transport paradigms:

- P2P
- BitTorrent
- RSS

Arbitrary metadata:

Metadata categories:

- hierarchy
- attribute – value pairs
- keywords
- links

Abstraction of data collections:

Disparate systems can look the same and share data:

- Source configuration (SCM) and content management (CMS)
  - CVS, Subversion, Arch, Git, etc.
- Package management:
  - Debian, Red Hat, Gentoo, Starkit, Solaris, etc.

Multiple functions can be stacked or daisy-chained:

Imagine all the multiple tasks of a software release – building, archiving, backup, packaging, uploading, mirroring, notification – each accomplished with its own VFS, all VFS's stacked together, and the process initiated with a single drag and drop.

Desktop integration:

Make Tcl virtual filesystems visible to the operating system, and thus to all programs.

We have the technology:

- FUSE on Linux
- WebDrive on Windows

Thank you!

Steve Huntley

- stephen.huntley@alum.mit.edu

virtual filesystem software and applications available at: http://filtr.sourceforge.net