

A Whistle Stop Tour of the Linux File System

Marcus D. Hanwell

Gentoo Linux Developer

Licence: BY-NC-SA 1 <http://creativecommons.org/licenses/by-nc-sa/1.0/>

12 November, 2005



Introduction

The Virtual File System

VFS Overview

File System Hierarchy

The Root of Everything

Filesystem Hierarchy Standard (FHS)

File System Hierarchy

Some Oddities & Changes

Linux Standard Base—All Your Boxes Are Belong to Us!

File System Management

Space Usage

Mounting File Systems

Permissions - Counting in Binary

File Permissions

Directory Permissions

Setting Permissions

Available File Systems

Major File Systems

Other File Systems

Increasing Security

I/O Schedulers

A Few Tips

Acknowledgements

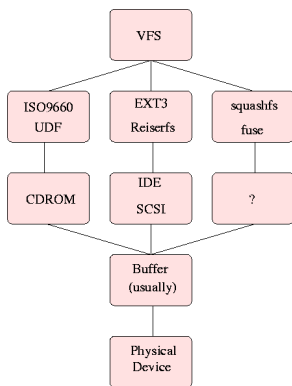


The Virtual File System

- ▶ Linux supports a large range of file systems
- ▶ This could lead to a large amount of duplicated code
- ▶ The Virtual File System is the answer!
 - ▶ Provides a standard interface to all file systems
 - ▶ Houses as much common code as possible
 - ▶ All file systems must use the VFS architecture



VFS Overview



- ▶ The VFS provides a level of separation from the raw file system
- ▶ It provides a standard set of features common to all file systems
- ▶ The file systems then interface with device drivers which also provide a standard interface to the file systems
- ▶ Allows a wide range of file systems to be used efficiently
 - ▶ Virtual
 - ▶ Encrypted
 - ▶ Compressed
 - ▶ Physical



The Root of Everything

- ▶ The Linux file system all falls below '/'
- ▶ This gives a stable, scalable hierarchy insensitive to changes in physical devices
- ▶ All partitions, special devices, removeable media etc are mounted as children of '/'
- ▶ Different directories can reside on different physical devices, with different file systems
- ▶ Physical implementation is hidden



Filesystem Hierarchy Standard (FHS)

- ▶ There are many Linux distributions
- ▶ Need to standardise the file system layout
- ▶ The Filesystem Hierarchy Standard aims to do just that
- ▶ Most Linux distributions follow this standard to a greater or lesser degree
- ▶ Provides a more consistent target for developers to aim at
- ▶ Some aspects are less important than others



File System Hierarchy

- ▶ /boot—boot loader, kernel etc
- ▶ /bin—binary executables needed at boot
- ▶ /dev—device files (usually managed by devfs/udev)
- ▶ /etc—system configuration files
- ▶ /home—user files
- ▶ /lib—system libraries for binaries in /bin//sbin
- ▶ /media—removeable media managed by HAL/hotplug etc
- ▶ /mnt—mount points for static devices to be mounted



File System Hierarchy continued

- ▶ /opt—binary precompiled files provided by vendors
- ▶ /proc—pseudo file system resident in memory (kernel info)
- ▶ /root—the root user's home directory (usually under '/')
- ▶ /sbin—system binaries needed at boot - for use only by root usually
- ▶ /sys—managed by the new sysfs driver with similar attributes to /proc
- ▶ /tmp—temporary files (world writeable)



File System Hierarchy continued

- ▶ `/usr`—user files, executables etc
 - ▶ `bin`—user executables - most executables are in here
 - ▶ `doc`—system documentation
 - ▶ `include`—system include files
 - ▶ `lib`—libraries for executables
 - ▶ `local`—local binaries, libs etc compiled by the user/admin
 - ▶ `man`—man files for the system
 - ▶ `sbin`—system binaries for root use only (usually)
 - ▶ `share`—arch neutral supplementary files used by apps
 - ▶ `src`—source code for programs (linux usually)
- ▶ `/var`—variable files (changed a lot)
 - ▶ system logs
 - ▶ mail spools, etc



Some Oddities & Changes

- ▶ The X server is usually at `/usr/X11R6`, but this is changing—on Gentoo X11R6 is a symlink to `../usr`
- ▶ `/usr/kde/3.5` houses KDE 3.5.*, in this way multiple versions can be slotted on one system
- ▶ Qt 3.* is in `/usr/qt/3`, but Qt 4 is in the normal file system hierarchy
- ▶ These moves have advantages (being in the normal hierarchy) but it does make it harder to slot multiple versions



Linux Standard Base—All Your Boxes Are Belong to Us!

- ▶ Based on the assumption that all distros are Red Hat derivatives!
- ▶ Aims at full binary compatibility across all distros
- ▶ Surprisingly Red Hat are spear heading this “standardisation” initiative. . .
- ▶ Totally inappropriate for source based distros
- ▶ Not even that appropriate for Red Hat derived distros
- ▶ It has even been criticised by Red Hat employees



Space Usage

There are a large range of command line and GUI applications. The `-h` switch gives sizes in human readable form (for those of us who can't do it in our heads!)

- ▶ `du` to check how much space directories and files consume
 - ▶ `du -sh /dir` gives the space used by `/dir` and all its contents
- ▶ `df` to check file system space usage
 - ▶ `df -h` lists the space used on all mounted volumes



Mounting File Systems

The `mount` command is used to mount file systems in Linux. One very important configuration file is `/etc/fstab` which tells the system how to mount all the important file systems on a Linux system.

```
/dev/hda1 /boot ext2 noauto 1 2
```

With the above entry you can simply type `# mount /boot` and it will use the options in `fstab`, otherwise entering the full form `# mount -t ext2 /dev/hda1 /boot` will accomplish the same thing. Either way `# umount /boot` unmounts it.



Permissions - Counting in Binary

- ▶ Now you need to learn to count to 7 in binary!

Number	Read	Write	Execute
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



File Permissions

- ▶ Hopefully those numbers make a little more sense now
- ▶ Files and directories have 3 permissions - owner, group and world as well as special bits for setuid, setgid etc
- ▶ When applied to files these permissions mean
 - ▶ read—can be read, but not changed
 - ▶ write—can be written to
 - ▶ execute—can be executed
- ▶ There are a few special file permissions
 - ▶ Set user ID—`--rwsr-xr-x` executed as the file owner
 - ▶ Set group ID—`--rwxr-sr-x` executed as the group
 - ▶ Both attributes are a potential security risk



Directory Permissions

- ▶ When applied to directories these permissions mean
 - ▶ read—contents can be read and displayed
 - ▶ write—contents can be changed
 - ▶ execute—can be entered
- ▶ There are two special directory permissions too
 - ▶ Set group ID—`drwxrwsrwx` where all files saved in this directory are owned by that group
 - ▶ Sticky bit—`drwxrwxrwt` only the user that created the file/dir can delete it



Setting Permissions

- ▶ The `chmod` command is used to set file/directory permissions using octal or symbolic syntax
 - ▶ `chmod 755 test`—`--rwxr-xr-x`
 - ▶ `chmod g+w test`—`--rwxrwxr-x`
 - ▶ `chmod u+s test`—`--rwsrwxr-x`
- ▶ The `chown` and `chgrp` commands change the file owner or group
 - ▶ `chown marcus:gentoo test` would change the file owner to the user `marcus` and the group `gentoo`



Major File Systems

There is no ultimate file system, it depends up on *your* useage

ext2 Tried and tested—the most tested fs available for Linux

ext3 ext2 with added journalling support, reliable and well tested

reiserfs Fast for lots of small files, with journalling

resier4 Even faster than reiserfs but still has bugs—needs more testing

jfs Journalled file system from IBM optimised for throughput

xfs Good performance for lots of large files



Other File Systems

- ▶ The device mapper offers many powerful features
- ▶ Software RAID and LVM2 offer many more options
- ▶ On the fly encrypted file systems for increased security—typically /home, /tmp and swap on laptop systems
- ▶ User space file systems are a big thing, FUSE just made it into the 2.6.14 kernel!
 - ▶ sshfs allows a remote file system to be mounted over ssh
 - ▶ encfs provides an encrypted file system
 - ▶ Fusedav provides access to webDAV repos
 - ▶ GmailFS uses gmail as a remote file system
 - ▶ Even WikipediaFS is available amongst others!



Increasing Security

- ▶ A few mount options that can increase system security
- ▶ The `nosuid` option ignores the `suid` bit
- ▶ The `noexec` option prevents anything being executed
- ▶ The `nodev` option ignores any device files, all but `/` can safely be mounted with this option

```
/dev/sda5 /tmp reiserfs notail,noatime,nodev,nosuid,noexec
```

- ▶ You can even mount partitions like `/usr` read only
 - ▶ Remount when updating/installing apps



I/O Schedulers

- ▶ Anticipatory
 - ▶ Default scheduler
 - ▶ Delays before I/O for possible reorders
 - ▶ Designed for slow storage
- ▶ Deadline
 - ▶ Reorders disk I/O like anticipatory to reduce head movement
 - ▶ Introduces a deadline to prevent resource starvation
- ▶ CFQ
 - ▶ Tries to distribute bandwidth equally
 - ▶ Suitable for most desktop systems



- ▶ /boot doesn't need a journalling file system, ext2 is fine
- ▶ /boot doesn't even need to be mounted normally. . .
- ▶ /var and /tmp benefit most from reiserfs
- ▶ /, /home, /usr probably benefit most from ext3 as it offers the best data integrity assurance
- ▶ RAID 1 constantly mirrors your data across two or more drives
- ▶ RAID 5 mirrors and stripes your data (3+ drives)
 - ▶ Even with RAID 5 regular backups need to be made—it cannot prevent data corruption (but will back it up)
- ▶ jfs and xfs are not really aimed at normal users
- ▶ UPSes are great for data integrity, as is good RAM!



Acknowledgements

- ▶ I would like to thank you all for listening
- ▶ This presentation was created in \LaTeX using the beamer class
- ▶ Linux is all about choice, and I have only had time to show a few of them to you...

