

A streamlined and fully compatible Linux Filesystem Hierarchy

Kay Sievers <kay@redhat.com>
Harald Hoyer <harald@redhat.com>

```
/
|-- dev
|-- etc
|-- proc
|-- run
|-- tmp
|-- sys
|-- usr
    |-- bin
    `-- lib
-- var
    |-- cache
    |-- lib
    |-- log
    `-- tmp
```

etc - host only configuration (possibly ro)

var - host only state/data (rw)

usr - system/distribution (ro by default, shareable)

```
| -- boot
| -- dev
| -- etc
| -- home
| -- mnt
| -- opt
| -- proc
| -- root
| -- run
| -- srv
| -- tmp
| -- sys
| -- usr
|   |-- bin
|   |-- include
|   |-- lib
|   |-- local
|   |-- share
|-- var
|   |-- cache
|   |-- lib
|   |-- log
|   |-- tmp
```

```
/
|-- bin → usr/bin
|-- boot
|-- dev
|-- etc
|-- home
|-- lib → usr/lib
|-- lib64 → usr/lib/x86-64-linux-gnu
|-- mnt
|-- opt
|-- proc
|-- root
|-- run
|-- sbin → usr/bin
|-- srv
|-- tmp
|-- sys
|-- usr
|   |-- bin
|   |-- include
|   |-- lib
|       |-- i386-linux-gnu
|           |-- ld-linux.so.2 → ld-2.15.so
|           |-- ld-linux.so.2 → i386-linux-gnu/ld-linux.so.2
|       `-- x86_64-linux-gnu
|           |-- ld-linux-x86-64.so.2 → ld-2.15.so
|-- lib64 → lib/x86_64-linux-gnu
|-- local
|-- sbin → bin
`-- share
-- var
    |-- cache
    |-- lib
    |-- lock → ../run/lock
    |-- log
    |-- run → ../run
    `-- tmp
```

Multi-lib

- future-proof multilib, tri/quad-arch layout
- strict separation of shared libraries and other lib directory content

usr

```
|-- lib
|   |-- i386-linux-gnu
|   |   |-- ld-2.15.so
|   |   |-- ld-linux.so.2 → ld-2.15.so
|   |   |-- libc-2.15.so
|   |   `-- libc.so.6 → libc-2.15.so
|   |-- ld-linux.so.2 → i386-linux-gnu/ld-linux.so.2
|   |-- x86_64-linux-gnu
|   |   |-- ld-2.15.so
|   |   |-- ld-linux-x86_64.so.2 → ld-2.15.so
|   |   |-- libc-2.15.so
|   |   `-- libc.so.6 → libc-2.15.so
|   `-- x86_x32-linux-gnu
|       |-- ld-2.15.so
|       |-- ld-linux-x86_x32.so.2 → ld-2.15.so
|       |-- libc-2.15.so
|       `-- libc.so.6 → libc-2.15.so
|-- lib64 → lib/x86_64-linux-gnu
```

/sbin-move

- single directory for executables in \$PATH
- best possible compatibility with other Linux distributions or UNIX variants
- daemons should not live in /usr/sbin, because they should not be started from an interactive shell anyway
- usermode / consolehelper should be removed and replaced by PolicyKit (pkexec)
- capabilities and security modules blur the distinction between root and non-root

```
/
|-- bin → usr/bin
|-- sbin → usr/bin
|-- usr
    |-- bin
    `-- sbin → bin
```

Application private directories

- phase out the Fedora-only libexec directory
- do not use /usr/share for things which are not shared between packages

```
/usr/lib/udev/  
|-- accelerometer  
|-- ata_id  
|-- bluetooth_serial  
|-- cdrom_id  
...  
|-- rules.d  
|   |-- 10-dm.rules  
...  
|   `-- 99-systemd.rules  
|-- scsi_id  
|-- tascam_fpga  
|-- tascam_fw  
|-- udevd  
...  
`-- v4l_id
```

```
/usr/lib/<package-name>/
```

LSB:

"Applications may use a single subdirectory under /usr/lib. If an application uses a subdirectory, all data exclusively used by the application must be placed within that subdirectory."

udev does **not** belong in /usr/sbin!

Do not mess with /boot via RPMs

```
/usr/lib/modules/<kernel-version>/  
build/  
extra/  
kernel/  
updates/  
modules.*  
vmlinuz  
System.map
```

- **copy vmlinuz to /boot with** new-kernel-pkg
- **remove old files with** new-kernel-pkg (old-kernel-pkg?? :-)
- **initramfs created in /boot with modules found in**
/usr/lib/modules/<kernel-version>/
- /boot is owned by the machine, and not by the distribution

/etc

- Should only be configured with host specific configuration data
- Default configuration files deployed in rpm packages should live in `/usr/lib/<package-name>` or be compiled in.

Example: systemd and udev configuration file overload

- `/etc/udev/rules.d/<foo>.rules` **overrides shipped**
`/usr/lib/udev/rules.d/<foo>.rules`
- `/etc/udev/rules.d/<foo>.rules` **symlink to /dev/null disables shipped** `<foo>.rules`

/tmp vs /var/tmp

/tmp

- by default a 'tmpfs' filesystem
- 'small' temporary files
- not preserved between system reboots
- automatic time-based clean-up

/var/tmp

- needed for 'large' files, which might not fit into a tmpfs /tmp
- temporary files preserved between system reboots
- automatic time-based clean-up

Thanks for Listening

Links:

- <http://wiki.debian.org/Multiarch/Tuples>