

# virtiofs: virtio-fs host<->guest shared file system

- Copyright (C) 2019 Red Hat, Inc.

## Introduction

The virtiofs file system for Linux implements a driver for the paravirtualized VIRTIO "virtio-fs" device for guest<->host file system sharing. It allows a guest to mount a directory that has been exported on the host.

Guests often require access to files residing on the host or remote systems. Use cases include making files available to new guests during installation, booting from a root file system located on the host, persistent storage for stateless or ephemeral guests, and sharing a directory between guests.

Although it is possible to use existing network file systems for some of these tasks, they require configuration steps that are hard to automate and they expose the storage network to the guest. The virtio-fs device was designed to solve these problems by providing file system access without networking.

Furthermore the virtio-fs device takes advantage of the co-location of the guest and host to increase performance and provide semantics that are not possible with network file systems.

## Usage

Mount file system with tag `myfs` on `/mnt`:

```
guest# mount -t virtiofs myfs /mnt
```

Please see <https://virtio-fs.gitlab.io/> for details on how to configure QEMU and the virtiofsd daemon.

## Mount options

virtiofs supports general VFS mount options, for example, `remount`, `ro`, `rw`, `context`, etc. It also supports FUSE mount options.

### atime behavior

The atime-related mount options, for example, `noatime`, `strictatime`, are ignored. The atime behavior for virtiofs is the same as the underlying filesystem of the directory that has been exported on the host.

## Internals

Since the virtio-fs device uses the FUSE protocol for file system requests, the virtiofs file system for Linux is integrated closely with the FUSE file system client. The guest acts as the FUSE client while the host acts as the FUSE server. The `/dev/fuse` interface between the kernel and userspace is replaced with the virtio-fs device interface.

FUSE requests are placed into a virtqueue and processed by the host. The response portion of the buffer is filled in by the host and the guest handles the request completion.

Mapping `/dev/fuse` to virtqueues requires solving differences in semantics between `/dev/fuse` and virtqueues. Each time the `/dev/fuse` device is read, the FUSE client may choose which request to transfer, making it possible to prioritize certain requests over others. Virtqueues have queue semantics and it is not possible to change the order of requests that have been enqueued. This is especially important if the virtqueue becomes full since it is then impossible to add high priority requests. In order to address this difference, the virtio-fs device uses a "hiprio" virtqueue specifically for requests that have priority over normal requests.