

Fuse I/O Modes

Fuse supports the following I/O modes:

- direct-io
- cached + write-through + writeback-cache

The direct-io mode can be selected with the `FOPEN_DIRECT_IO` flag in the `FUSE_OPEN` reply.

In direct-io mode the page cache is completely bypassed for reads and writes. No read-ahead takes place. Shared mmap is disabled.

In cached mode reads may be satisfied from the page cache, and data may be read-ahead by the kernel to fill the cache. The cache is always kept consistent after any writes to the file. All mmap modes are supported.

The cached mode has two sub modes controlling how writes are handled. The write-through mode is the default and is supported on all kernels. The writeback-cache mode may be selected by the `FUSE_WRITEBACK_CACHE` flag in the `FUSE_INIT` reply.

In write-through mode each write is immediately sent to userspace as one or more `WRITE` requests, as well as updating any cached pages (and caching previously uncached, but fully written pages). No `READ` requests are ever sent for writes, so when an uncached page is partially written, the page is discarded.

In writeback-cache mode (enabled by the `FUSE_WRITEBACK_CACHE` flag) writes go to the cache only, which means that the `write(2)` syscall can often complete very fast. Dirty pages are written back implicitly (background writeback or page reclaim on memory pressure) or explicitly (invoked by `close(2)`, `fsync(2)` and when the last ref to the file is being released on `munmap(2)`). This mode assumes that all changes to the filesystem go through the FUSE kernel module (size and `atime/ctime/mtime` attributes are kept up-to-date by the kernel), so it's generally not suitable for network filesystems. If a partial page is written, then the page needs to be first read from userspace. This means, that even for files opened for `O_WRONLY` it is possible that `READ` requests will be generated by the kernel.