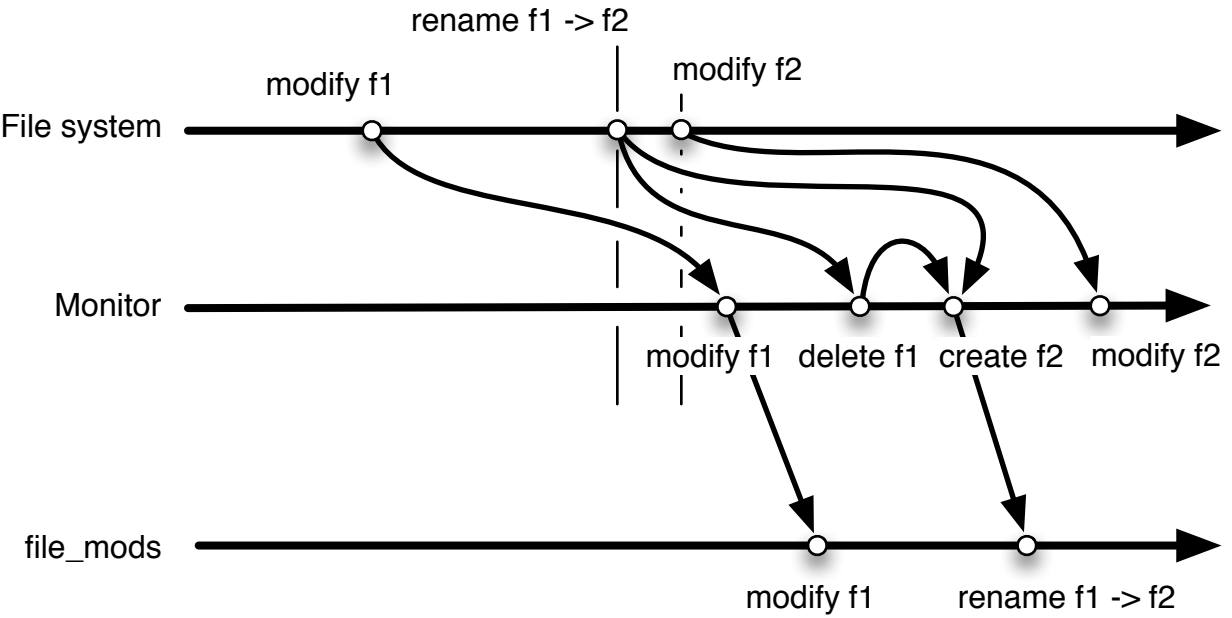

This diagram illustrates two pathological cases that will cause us problems in the state transformations in the monitor.

The first occurs with just the modification of file *f1* followed by a rename *f1* to *f2*. If the FileWatcher event describing the modification of *f1* is delivered after the rename is completed, then the monitor will not be able to read the contents of *f1* in order to send the correct modification event with the changed values to the consumer. What needs to happen is that we need to process the modify, delete and create events as a single entity so that we can guess that we should look at f2 for the changes and then emit the changes as if they were applied to *f1.*

This raises another problem. Suppose that a modification is made to *f2* just after the rename, but before the first modification operation is seen. As before the modify, delete and create will be understood as a single entity, but when *f2* is read to find out what changed, the changes both before and after the rename will be detected. These changes will erroneously be emitted before the rename.

There isn't much we can do about this. Since the first idiom (change, rename, no more changes) is very common and the second idiom (change, rename, more changes) is rare, we should probably not worry about this corner case. We should test both scenarios, however.