File Events

`FileWatching.poll_fd` — Function

```
poll_fd(fd, timeout_s::Real=-1; readable=false, writable=false)
```

Monitor a file descriptor `fd` for changes in the read or write availability, and with a timeout given by `timeout_s` seconds.

The keyword arguments determine which of read and/or write status should be monitored; at least one of them must be set to `true`.

The returned value is an object with boolean fields `readable`, `writable`, and `timedout`, giving the result of the polling.

`FileWatching.poll_file` — Function

```
poll_file(path::AbstractString, interval_s::Real=5.007, timeout_s::Real=-1) ->
```

Monitor a file for changes by polling every `interval_s` seconds until a change occurs or `timeout_s` seconds have elapsed. The `interval_s` should be a long period; the default is 5.007 seconds.

Returns a pair of status objects `(previous, current)` when a change is detected. The `previous` status is always a `StatStruct`, but it may have all of the fields zeroed (indicating the file didn't previously exist, or wasn't previously accessible).

The `current` status object may be a `StatStruct`, an `EOFError` (indicating the timeout elapsed), or some other `Exception` subtype (if the `stat` operation failed - for example, if the path does not exist).

To determine when a file was modified, compare `current isa StatStruct && mtime(prev) != mtime(current)` to detect notification of changes. However, using `watch_file` for this operation is preferred, since it is more reliable and efficient, although in some situations it may not be available.

`FileWatching.watch_file` — Function

```
watch_file(path::AbstractString, timeout_s::Real=-1)
```

Watch file or directory `path` for changes until a change occurs or `timeout_s` seconds have elapsed.

The returned value is an object with boolean fields `changed`, `renamed`, and `timedout`, giving the result of watching the file.

This behavior of this function varies slightly across platforms. See https://nodejs.org/api/fs.html#fs_caveats for more detailed information.

`FileWatching.watch_folder` — Function

```
watch_folder(path::AbstractString, timeout_s::Real=-1)
```

Watches a file or directory `path` for changes until a change has occurred or `timeout_s` seconds have elapsed.

This will continuing tracking changes for `path` in the background until `unwatch_folder` is called on the same `path`.

The returned value is an pair where the first field is the name of the changed file (if available) and the second field is an object with boolean fields `changed`, `renamed`, and `timedout`, giving the event.

This behavior of this function varies slightly across platforms. See https://nodejs.org/api/fs.html#fs_caveats for more detailed information.

`FileWatching.unwatch_folder` — Function

```
unwatch_folder(path::AbstractString)
```

Stop background tracking of changes for `path`. It is not recommended to do this while another task is waiting for `watch_folder` to return on the same path, as the result may be unpredictable.

« Distributed Computing Future »

Powered by Documenter.jl and the Julia Programming Language.