
pyfanotify

Release 0.2.2

baskiton

Dec 16, 2023

CONTENTS:

1	Usage	3
2	Contents	5
2.1	Installation	5
2.2	pyfanotify	5
3	Indices and tables	9
	Python Module Index	11
	Index	13

pyfanotify - is a Python wrapper for the Linux fanotify.

See also:

[man fanotify](#) for more info on fanotify.

Note: Required python 3.6+

<p>Warning: Required execution from ROOT</p>
--

CHAPTER
ONE

USAGE

See [examples](#)

CONTENTS

2.1 Installation

2.1.1 Using PIP

```
$ pip install pyfanotify
```

2.1.2 From sources

```
$ git clone https://github.com/baskiton/pyfanotify.git
$ cd pyfanotify
$ python setup.py install
```

2.2 pyfanotify

Note: Requires execution from **ROOT**!

class pyfanotify.Fanotify

Wrapper for Linux fanotify. Runs in a new process.

__init__ (*init_fid=False, log=None, fn=None, fn_args=(), fn_timeout=0*)

Parameters

- **init_fid** (*bool*) – Enable filesystem events to watch (FAN_CREATE, FAN_DELETE, FAN_MOVE, FAN_ATTRIB). See **man fanotify_init** for FAN_REPORT_FID and FAN_REPORT_DIR_FID
- **log** (*Optional[Logger]*) – Logger
- **fn** (*Optional[Callable]*) – Function that will be called after the specified *fn_timeout*
- **fn_args** (*Tuple*) – Arguments for *fn*
- **fn_timeout** (*int*) – Timeout for *fn*

Raises

- **OSError** – if fanotify is not set in kernel or other fanotify error (see **man fanotify_init**)

- **TypeError** – if *fn* is not callable or *fn_args* is not tuple

property with_fid: bool

start()

Start Fanotify process

Return type

None

stop()

Stop Fanotify process

Return type

None

connect(*rule*)

Add [FanoRule](#) to receive events on it

Parameters

rule ([FanoRule](#)) –

Return type

None

disconnect(*rule*)

Delete the [FanoRule](#) so as not to receive events for it

Parameters

rule ([FanoRule](#)) –

Return type

None

mark(*path*, *ev_types*=FAN_ALL_EVENTS, *is_type*="", *dont_follow*=False, *as_ignore*=False, *remove*=False)

To detail see **man fanotify_mark**

Adds, removes, or modifies an fanotify mark on a filesystem object. The caller must have read permission on the filesystem object that is to be marked. *ev_types* must be nonempty

Parameters

- **path** (*Iterable*) – path to be marked
- **ev_types** (*int*) – defines which events shall be listened for (or which shall be ignored). It is a bit mask composed values. See **man**
- **is_type** (*str*) – type of *path*. It can be:
 - 'mp' - is mount point
 - 'fs' - is filesystem
 - 'dir' - is directory
- **dont_follow** (*bool*) – if *path* is a symbolic link, mark the link itself, rather than the file to which it refers.
- **as_ignore** (*bool*) – if *True* add/remove to/from ignore mask.
- **remove** (*bool*) – if *True*, events in *ev_types* will be removed from the mark mask (or from ignore mask); else events will be added to the mark mask (or to ignore mask).

Return type

None

flush(*do_non_mounts=True, do_mounts=True, do_fs=True*)

To detail see **man fanotify_mark** for FAN_MARK_FLUSH

Remove either all marks for filesystems, all marks for mounts, or all marks for directories and files from the fanotify group.

Parameters

- **do_non_mounts** – Remove all marks for directories and files
- **do_mounts** – Remove all marks for mounts
- **do_fs** – Remove all marks for filesystems (since Linux 4.20)

Return type

None

class pyfanotify.FanoRule

FanoRule(name, pids=(), ev_types=0, exe_pattern=None, cwd_pattern=None, path_pattern=None, pass_fd=False)

Rule to receive events on it via fanotify. At least one rule parameter must be specified (other than the required *name* and optional *pass_fd*)

Parameters

- **name** (*AnyStr*) – Name of rule
- **pids** (*Iterable[Union[int, AnyStr]]*) – PIDS
- **ev_types** (*int*) – Event types mask
- **exe_pattern** (*AnyStr*) – exe
- **cwd_pattern** (*AnyStr*) – cwd
- **path_pattern** (*AnyStr*) – path
- **pass_fd** (*bool*) – Pass file descriptor

Raises

- **TypeError** – if *pids* is not a set, list or tuple
- **ValueError** – if no one rule parameter are specified

class pyfanotify.FanotifyClient

Client for easy use and getting data via Fanotify.

__init__(*fanotify, **rkw*)

Parameters

- **fanotify** (*Fanotify*) – *Fanotify* object to associate with it.
- **rkw** (*dict*) – Keyword arguments for *FanoRule*, excluding *FanoRule.name* - this will be auto-generated and stored to *FanotifyClient.rname*

Return type

None

close()

Close the connection to the Fanotify object. The data will no longer be received.

Return type

None

get_events()

Receive fanotify events according to the established rules for the current client.

Return type

[FanotifyData](#)

class pyfanotify.FanotifyData

Contains fanotify event information

fd: **int** = -1

File descriptor if passed, -1 otherwise

pid: **int** = 0

PID of caused process

ev_types: **int** = 0

Event types of fanotify event

exe: **str** = None

EXE of the event caused process if passed, None otherwise

cwd: **str** = None

CWD of the event caused process if passed, None otherwise

path: **str** = None

PATH of the event caused file if passed, None otherwise

__init__(*fd=-1, pid=0, ev_types=0, exe=None, cwd=None, path=None*)

Parameters

- **fd** (*int*) – File descriptor if passed
- **pid** (*int*) – PID of caused process
- **ev_types** (*int*) – Event types of fanotify event
- **exe** (*Optional[str]*) – EXE of the event caused process if passed
- **cwd** (*Optional[str]*) – CWD of the event caused process if passed
- **path** (*Optional[str]*) – PATH of the event caused file if passed

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

p

`pyfanotify` (*Linux*), 5

Symbols

`__init__()` (*pyfanotify.Fanotify method*), 5
`__init__()` (*pyfanotify.FanotifyClient method*), 7
`__init__()` (*pyfanotify.FanotifyData method*), 8

C

`close()` (*pyfanotify.FanotifyClient method*), 7
`connect()` (*pyfanotify.Fanotify method*), 6
`cwd` (*pyfanotify.FanotifyData attribute*), 8

D

`disconnect()` (*pyfanotify.Fanotify method*), 6

E

`ev_types` (*pyfanotify.FanotifyData attribute*), 8
`exe` (*pyfanotify.FanotifyData attribute*), 8

F

`FanoRule` (*class in pyfanotify*), 7
`Fanotify` (*class in pyfanotify*), 5
`FanotifyClient` (*class in pyfanotify*), 7
`FanotifyData` (*class in pyfanotify*), 8
`fd` (*pyfanotify.FanotifyData attribute*), 8
`flush()` (*pyfanotify.Fanotify method*), 6

G

`get_events()` (*pyfanotify.FanotifyClient method*), 7

M

`mark()` (*pyfanotify.Fanotify method*), 6
`module`
 pyfanotify, 5

P

`path` (*pyfanotify.FanotifyData attribute*), 8
`pid` (*pyfanotify.FanotifyData attribute*), 8
`pyfanotify`
 module, 5

S

`start()` (*pyfanotify.Fanotify method*), 6

`stop()` (*pyfanotify.Fanotify method*), 6

W

`with_fid` (*pyfanotify.Fanotify property*), 6