

Unittest for fstools

August 15, 2025

Contents

1	Test Information	3
1.1	Test Candidate Information	3
1.2	Unittest Information	3
1.3	Test System Information	3
2	Statistic	3
2.1	Test-Statistic for testrun with python 3.13.5 (final)	3
2.2	Coverage Statistic	4
3	Testcases with no corresponding Requirement	5
3.1	Summary for testrun with python 3.13.5 (final)	5
3.1.1	Test creating a folder with fstools.mkdir	5
3.1.2	Test creating a folder with fstools.mkdir, when target exists	5
3.1.3	Test for fstools.dirlist (not existing folder)	6
3.1.4	Test for fstools.dirlist (not rekursive)	6
3.1.5	Test for fstools.dirlist (rekursive)	6
3.1.6	Test for fstools.filelist (not existing folder)	7
3.1.7	Test for fstools.filelist (not rekursive; all files)	7
3.1.8	Test for fstools.filelist (not rekursive; files ending with .rw)	8
3.1.9	Test for fstools.filelist (rekursive; all files)	8
3.1.10	Test for fstools.filelist (rekursive; files ending with .ro)	8
3.1.11	Test for fstools.is_writable	9
3.1.12	Test for fstools.openlocked_blocking	9
3.1.13	Test for fstools.openlocked_non_blocking	10
A	Trace for testrun with python 3.13.5 (final)	11
A.1	Tests with status Info (13)	11
A.1.1	Test creating a folder with fstools.mkdir	11
A.1.2	Test creating a folder with fstools.mkdir, when target exists	12
A.1.3	Test for fstools.is_writable	13
A.1.4	Test for fstools.filelist (rekursive; all files)	14

A.1.5	Test for fstools.filelist (rekursive; files ending with .ro)	15
A.1.6	Test for fstools.filelist (not rekursive; all files)	16
A.1.7	Test for fstools.filelist (not rekursive; files ending with .rw)	17
A.1.8	Test for fstools.filelist (not existing folder)	18
A.1.9	Test for fstools.dirlist (rekursive)	18
A.1.10	Test for fstools.dirlist (not rekursive)	19
A.1.11	Test for fstools.dirlist (not existing folder)	20
A.1.12	Test for fstools.openlocked_non_blocking	21
A.1.13	Test for fstools.openlocked_blocking	22
B	Test-Coverage	22
B.1	fstools	22
B.1.1	fstools.__init__.py	22

1 Test Information

1.1 Test Candidate Information

The Module `fstools` is designed to help on all issues with files and folders. For more Information read the documentation.

Library Information	
Name	fstools
State	Released
Supported Interpreters	python3
Version	54da6f4b94010fba1501b316e7e91e50
Dependencies	

1.2 Unittest Information

Unittest Information	
Version	d1a7a5c9b13c843a9e45b4fbcc505c9b
Testruns with	python 3.13.5 (final)

1.3 Test System Information

System Information	
Architecture	64bit
Distribution	Debian GNU/Linux 13 trixie
Hostname	ahorn
Kernel	6.12.38+deb13-amd64 (#1 SMP PREEMPT_DYNAMIC Debian 6.12.38-1 (2025-07-16))
Machine	x86_64
Path	/home/dirk/work/unittest_collection/fstools
System	Linux
Username	dirk

2 Statistic

2.1 Test-Statistic for testrun with python 3.13.5 (final)

Number of tests	13
Number of successfull tests	13
Number of possibly failed tests	0
Number of failed tests	0
Executionlevel	Full Test (all defined tests)
Time consumption	0.546s

2.2 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
fstools	64.4%	60.0%
fstools.__init__.py	64.4%	

3 Testcases with no corresponding Requirement

3.1 Summary for testrun with python 3.13.5 (final)

3.1.1 Test creating a folder with fstools.mkdir

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.1!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/report/_init_.py (331)
Start-Time:	2025-08-15 19:37:40,790
Finished-Time:	2025-08-15 19:37:40,794
Time-Consumption	0.004s
Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	fstools.mkdir executed to creating folder .../mkdir_test_folder.1
Success	Returnvalue of fstools.mkdir is correct (Content True and Type is <class 'bool'>).
Success	Returvalue of os.path.isdir is correct (Content True and Type is <class 'bool'>).
Info	fstools.mkdir executed to creating folder .../mkdir_test_folder.2/sub_folder
Success	Returnvalue of fstools.mkdir is correct (Content True and Type is <class 'bool'>).
Success	Returvalue of os.path.isdir is correct (Content True and Type is <class 'bool'>).
Info	fstools.mkdir executed to creating folder .../mkdir_test_folder.3
Success	Returnvalue of fstools.mkdir is correct (Content True and Type is <class 'bool'>).
Success	Returvalue of os.path.isdir is correct (Content True and Type is <class 'bool'>).

3.1.2 Test creating a folder with fstools.mkdir, when target exists

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.2!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/report/_init_.py (331)
Start-Time:	2025-08-15 19:37:40,794
Finished-Time:	2025-08-15 19:37:40,797
Time-Consumption	0.003s
Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	fstools.mkdir executed to creating folder .../writable_folder
Success	Returnvalue of fstools.mkdir is correct (Content True and Type is <class 'bool'>).
Info	fstools.mkdir executed to creating folder .../writable_file.rw
Success	Returnvalue of fstools.mkdir is correct (Content False and Type is <class 'bool'>).

3.1.3 Test for fstools.dirlist (not existing folder)

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.11!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unitest_collection/fstools/unitest/src/report/_init_.py (331)
Start-Time:	2025-08-15 19:37:40,821
Finished-Time:	2025-08-15 19:37:40,824
Time-Consumption	0.002s
Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder
Info	Executing dirlist to get list for /home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder/not_existing_folder
Success	Returnvalue of dirlist is correct (Content [] and Type is <class 'list'>).

3.1.4 Test for fstools.dirlist (not rekursive)

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.10!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unitest_collection/fstools/unitest/src/report/_init_.py (331)
Start-Time:	2025-08-15 19:37:40,819
Finished-Time:	2025-08-15 19:37:40,821
Time-Consumption	0.002s
Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder
Info	Executing dirlist to get list for /home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder
Success	Returnvalue of dirlist is correct (Content ['/home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder/writable_folder', '/home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder/not_writable_folder'] and Type is <class 'list'>).

3.1.5 Test for fstools.dirlist (rekursive)

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.9!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unitest_collection/fstools/unitest/src/report/_init_.py (331)
Start-Time:	2025-08-15 19:37:40,815
Finished-Time:	2025-08-15 19:37:40,819
Time-Consumption	0.003s

Testsummary:

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing dirlist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of dirlist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder'] and Type is <class 'list'>).

3.1.6 Test for fstools.filelist (not existing folder)**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.8!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/report/_init_.py (331)
Start-Time:	2025-08-15 19:37:40,813
Finished-Time:	2025-08-15 19:37:40,815
Time-Consumption	0.002s

Testsummary:

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_existing_folder
Success	Returnvalue of filelist is correct (Content [] and Type is <class 'list'>).

3.1.7 Test for fstools.filelist (not rekursive; all files)**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.6!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/report/_init_.py (331)
Start-Time:	2025-08-15 19:37:40,808
Finished-Time:	2025-08-15 19:37:40,810
Time-Consumption	0.002s

Testsummary:

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw'] and Type is <class 'list'>).

3.1.8 Test for fstools.filelist (not rekursive; files ending with .rw)

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.7!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/report/_init....py (331)
Start-Time:	2025-08-15 19:37:40,810
Finished-Time:	2025-08-15 19:37:40,813
Time-Consumption	0.002s

Testsummary:

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw'] and Type is <class 'list'>).

3.1.9 Test for fstools.filelist (rekursive; all files)

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.4!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/report/_init....py (331)
Start-Time:	2025-08-15 19:37:40,800
Finished-Time:	2025-08-15 19:37:40,804
Time-Consumption	0.004s

Testsummary:

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder/subfile.rw'] and Type is <class 'list'>).

3.1.10 Test for fstools.filelist (rekursive; files ending with .ro)

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.5!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/report/_init....py (331)

Start-Time: 2025-08-15 19:37:40,805
 Finished-Time: 2025-08-15 19:37:40,808
 Time-Consumption 0.003s

Testsummary:

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro'] and Type is <class 'list'>).

3.1.11 Test for fstools.is_writable**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.3!

Testrun: python 3.13.5 (final)
 Caller: /home/dirk/work/unittest_collection/fstools/unittest/src/report/_init_.py (331)
 Start-Time: 2025-08-15 19:37:40,797
 Finished-Time: 2025-08-15 19:37:40,800
 Time-Consumption 0.002s

Testsummary:

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Success Returnvalue of fstools.is_writable('.../writable_file.rw') is correct (Content True and Type is <class 'bool'>).
Success Returnvalue of fstools.is_writable('.../not_writable_file.ro') is correct (Content False and Type is <class 'bool'>).
Success Returnvalue of fstools.is_writable('.../writable_folder') is correct (Content True and Type is <class 'bool'>).
Success Returnvalue of fstools.is_writable('.../not_writable_folder') is correct (Content False and Type is <class 'bool'>).

3.1.12 Test for fstools.openlocked_blocking**Testresult**

This test was passed with the state: **Success**. See also full trace in section A.1.13!

Testrun: python 3.13.5 (final)
 Caller: /home/dirk/work/unittest_collection/fstools/unittest/src/report/_init_.py (331)
 Start-Time: 2025-08-15 19:37:40,831
 Finished-Time: 2025-08-15 19:37:41,338
 Time-Consumption 0.507s

Testsummary:

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Info	Opening a file with a blocking lock, writing to it, sleep a bit, trying to open and write to it from a child process
Success	Appended text to file is correct (Content 'Some more text.' and Type is <class 'str'>).

3.1.13 Test for fstools.openlocked_non_blocking

Testresult

This test was passed with the state: **Success**. See also full trace in section A.1.12!

Testrun:	python 3.13.5 (final)
Caller:	/home/dirk/work/unittest_collection/fstools/unittest/src/report/_init_.py (331)
Start-Time:	2025-08-15 19:37:40,824
Finished-Time:	2025-08-15 19:37:40,830
Time-Consumption	0.006s

Testsummary:

Info	Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Info	Opening a file with a non blocking lock, writing to it, trying to open and write to it from a child process and writing more text
Success	Exception name thrown in child process is correct (Content 'BlockingIOError' and Type is <class 'str'>).
Success	Appended text to file is correct (Content 'Some more text.' and Type is <class 'str'>).

A Trace for testrun with python 3.13.5 (final)

A.1 Tests with status Info (13)

A.1.1 Test creating a folder with fstools.mkdir

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating general Basefolder

↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating folder in general Basefolder: ../writable_folder

Creating folder in general Basefolder: ../not_writable_folder

Creating file in general Basefolder: ../writable_file.rw

Creating file in general Basefolder: ../not_writable_file.ro

Creating folder in general Basefolder: ../writable_folder/subfolder

Creating file in general Basefolder: ../writable_folder/subfolder/subfile.rw

Creating UID-Basefolder

↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder

Creating file in UID-Basefolder: ../uid_test_file

Creating folder in UID-Basefolder: ../uid_test_folder

Info fstools.mkdir executed to creating folder ../mkdir_test_folder_1

Success Returnvalue of fstools.mkdir is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of fstools.mkdir): True (<class 'bool'>)

Expectation (Returnvalue of fstools.mkdir): result = True (<class 'bool'>)

Success Returnvalue of os.path.isdir is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of os.path.isdir): True (<class 'bool'>)

Expectation (Returnvalue of os.path.isdir): result = True (<class 'bool'>)

Info fstools.mkdir executed to creating folder ../mkdir_test_folder_2/sub_folder

Success Returnvalue of fstools.mkdir is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of fstools.mkdir): True (<class 'bool'>)

Expectation (Returnvalue of fstools.mkdir): result = True (<class 'bool'>)

Success Returnvalue of os.path.isdir is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of os.path.isdir): True (<class 'bool'>)

Expectation (Returnvalue of os.path.isdir): result = True (<class 'bool'>)

Info fstools.mkdir executed to creating folder .../mkdir_test_folder_3

Success Returnvalue of fstools.mkdir is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of fstools.mkdir): True (<class 'bool'>)

Expectation (Returnvalue of fstools.mkdir): result = True (<class 'bool'>)

Success Returnvalue of os.path.isdir is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of os.path.isdir): True (<class 'bool'>)

Expectation (Returnvalue of os.path.isdir): result = True (<class 'bool'>)

A.1.2 Test creating a folder with fstools.mkdir, when target exists

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating general Basefolder

↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating folder in general Basefolder: .../writable_folder

Creating folder in general Basefolder: .../not_writable_folder

Creating file in general Basefolder: .../writable_file.rw

Creating file in general Basefolder: .../not_writable_file.ro

Creating folder in general Basefolder: .../writable_folder/subfolder

Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw

Creating UID-Basefolder

↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder

Creating file in UID-Basefolder: .../uid_test_file

Creating folder in UID-Basefolder: .../uid_test_folder

Info fstools.mkdir executed to creating folder .../writable_folder

Success Returnvalue of fstools.mkdir is correct (Content True and Type is <class 'bool'>).

Result (Returnvalue of fstools.mkdir): True (<class 'bool'>)

```
Expectation (Returnvalue of fstools.mkdir): result = True (<class 'bool'>)
```

Info fstools.mkdir executed to creating folder .../writable_file.rw

Success Returnvalue of fstools.mkdir is correct (Content False and Type is <class 'bool'>).

```
Result (Returnvalue of fstools.mkdir): False (<class 'bool'>)
```

```
Expectation (Returnvalue of fstools.mkdir): result = False (<class 'bool'>)
```

A.1.3 Test for fstools.is_writable

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
```

```
↳ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
```

```
Creating folder in general Basefolder: .../writable_folder
```

```
Creating folder in general Basefolder: .../not_writable_folder
```

```
Creating file in general Basefolder: .../writable_file.rw
```

```
Creating file in general Basefolder: .../not_writable_file.ro
```

```
Creating folder in general Basefolder: .../writable_folder/subfolder
```

```
Creating file in general Basefolder: .../writable_folder/subfolder/subfile.rw
```

```
Creating UID-Basefolder
```

```
↳ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
```

```
Creating file in UID-Basefolder: .../uid_test_file
```

```
Creating folder in UID-Basefolder: .../uid_test_folder
```

Success Returnvalue of fstools.is_writable('.../writable_file.rw') is correct (Content True and Type is <class 'bool'>).

```
Result (Returnvalue of fstools.is_writable(".../writable_file.rw")): True (<class 'bool'>)
```

```
Expectation (Returnvalue of fstools.is_writable(".../writable_file.rw")): result = True
↳ (<class 'bool'>)
```

Success Returnvalue of fstools.is_writable('.../not_writable_file.ro') is correct (Content False and Type is <class 'bool'>).

```
Result (Returnvalue of fstools.is_writable(".../not_writable_file.ro")): False (<class
↳ 'bool'>)
```

```
Expectation (Returnvalue of fstools.is_writable(".../not_writable_file.ro")): result = False
↳ (<class 'bool'>)
```

Success Returnvalue of `fstools.is_writable('.../writable_folder')` is correct (Content True and Type is `<class 'bool'>`).

Result (Returnvalue of `fstools.is_writable(".../writable_folder")`): True (`<class 'bool'>`)

Expectation (Returnvalue of `fstools.is_writable(".../writable_folder")`): result = True (`<class 'bool'>`)
 ↪ 'bool'>)

Success Returnvalue of `fstools.is_writable('.../not_writable_folder')` is correct (Content False and Type is `<class 'bool'>`).

Result (Returnvalue of `fstools.is_writable(".../not_writable_folder")`): False (`<class 'bool'>`)

Expectation (Returnvalue of `fstools.is_writable(".../not_writable_folder")`): result = False
 ↪ (`<class 'bool'>`)

A.1.4 Test for `fstools.filelist` (rekursive; all files)

Testresult

This test was passed with the state: **Success**.

Info Files and folders for `fstools` unittest created under `/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder`

Creating general Basefolder

↪ `/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder`

Creating folder in general Basefolder: `.../writable_folder`

Creating folder in general Basefolder: `.../not_writable_folder`

Creating file in general Basefolder: `.../writable_file.rw`

Creating file in general Basefolder: `.../not_writable_file.ro`

Creating folder in general Basefolder: `.../writable_folder/subfolder`

Creating file in general Basefolder: `.../writable_folder/subfolder/subfile.rw`

Creating UID-Basefolder

↪ `/home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder`

Creating file in UID-Basefolder: `.../uid_test_file`

Creating folder in UID-Basefolder: `.../uid_test_folder`

Info Executing `filelist` to get list for `/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder`

FILELIST: path (`/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder`)

↪ exists - looking for files to append

FILELIST: path (`/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/w_`)

↪ `ritable_folder`) exists - looking for files to append

FILELIST: path (`/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/w_`)

↪ `ritable_folder/subfolder`) exists - looking for files to append

```
FILELIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder) exists - looking for files to append
```

Success Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder/subfile.rw'] and Type is <class 'list'>).

```
Result (Returnvalue of filelist): [ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro',
↳ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw',
↳ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder/subfile.rw' ] (<class 'list'>)
```

```
Expectation (Returnvalue of filelist): result = [ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw',
↳ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro',
↳ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder/subfile.rw' ] (<class 'list'>)
```

```
Content '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro' is incorrect for test_variable.
```

A.1.5 Test for fstools.filelist (rekursive; files ending with .ro)

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating general Basefolder

```
↳ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
```

Creating folder in general Basefolder: ../writable_folder

Creating folder in general Basefolder: ../not_writable_folder

Creating file in general Basefolder: ../writable_file.rw

Creating file in general Basefolder: ../not_writable_file.ro

Creating folder in general Basefolder: ../writable_folder/subfolder

Creating file in general Basefolder: ../writable_folder/subfolder/subfile.rw

Creating UID-Basefolder

```
↳ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
```

Creating file in UID-Basefolder: ../uid_test_file

Creating folder in UID-Basefolder: ../uid_test_folder

Info Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
FILELIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder) exists - looking for files to append
```

```
FILELIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder) exists - looking for files to append
FILELIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder) exists - looking for files to append
FILELIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder) exists - looking for files to append
```

Success Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro'] and Type is <class 'list'>).

```
Result (Returnvalue of filelist): [ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro' ] (<class 'list'>)
Expectation (Returnvalue of filelist): result = [ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro' ] (<class 'list'>)
```

A.1.6 Test for fstools.filelist (not rekursive; all files)

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
Creating general Basefolder
↳ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: ../writable_folder
Creating folder in general Basefolder: ../not_writable_folder
Creating file in general Basefolder: ../writable_file.rw
Creating file in general Basefolder: ../not_writable_file.ro
Creating folder in general Basefolder: ../writable_folder/subfolder
Creating file in general Basefolder: ../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↳ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating file in UID-Basefolder: ../uid_test_file
Creating folder in UID-Basefolder: ../uid_test_folder
```

Info Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```
FILELIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder) exists - looking for files to append
```

Success Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw'] and Type is <class 'list'>).

```
Result (Returnvalue of filelist): [ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_file.ro',
↳ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw' ] (<class 'list'>)
```

```
Expectation (Returnvalue of filelist): result = [ '/home/dirk/work/unittest_collection/fstool
↳ s/unittest/output_data/testfolder/writable_file.rw',
↳ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_
↳ _file.ro' ] (<class 'list'>)
```

```
Content '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writ
↳ able_file.ro' is incorrect for test_variable.
```

A.1.7 Test for fstools.filelist (not rekursive; files ending with .rw)

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating general Basefolder

```
↳ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
```

Creating folder in general Basefolder: ../writable_folder

Creating folder in general Basefolder: ../not_writable_folder

Creating file in general Basefolder: ../writable_file.rw

Creating file in general Basefolder: ../not_writable_file.ro

Creating folder in general Basefolder: ../writable_folder/subfolder

Creating file in general Basefolder: ../writable_folder/subfolder/subfile.rw

Creating UID-Basefolder

```
↳ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
```

Creating file in UID-Basefolder: ../uid_test_file

Creating folder in UID-Basefolder: ../uid_test_folder

Info Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

FILELIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder)

```
↳ exists - looking for files to append
```

Success Returnvalue of filelist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_file.rw'] and Type is <class 'list'>).

```
Result (Returnvalue of filelist): [ '/home/dirk/work/unittest_collection/fstools/unittest/out
↳ put_data/testfolder/writable_file.rw' ] (<class 'list'>)
```

```
Expectation (Returnvalue of filelist): result = [ '/home/dirk/work/unittest_collection/fstool
↳ s/unittest/output_data/testfolder/writable_file.rw' ] (<class 'list'>)
```

A.1.8 Test for fstools.filelist (not existing folder)

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating general Basefolder

↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating folder in general Basefolder: ../writable_folder

Creating folder in general Basefolder: ../not_writable_folder

Creating file in general Basefolder: ../writable_file.rw

Creating file in general Basefolder: ../not_writable_file.ro

Creating folder in general Basefolder: ../writable_folder/subfolder

Creating file in general Basefolder: ../writable_folder/subfolder/subfile.rw

Creating UID-Basefolder

↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder

Creating file in UID-Basefolder: ../uid_test_file

Creating folder in UID-Basefolder: ../uid_test_folder

Info Executing filelist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_existing_folder

FILELIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_existing_folder) does not exist - empty filelist will be returned

Success Returnvalue of filelist is correct (Content [] and Type is <class 'list'>).

Result (Returnvalue of filelist): [] (<class 'list'>)

Expectation (Returnvalue of filelist): result = [] (<class 'list'>)

A.1.9 Test for fstools.dirlist (rekursive)

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating general Basefolder

↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating folder in general Basefolder: ../writable_folder

Creating folder in general Basefolder: ../not_writable_folder

Creating file in general Basefolder: ../writable_file.rw

Creating file in general Basefolder: ../not_writable_file.ro

Creating folder in general Basefolder: ../writable_folder/subfolder

Creating file in general Basefolder: ../writable_folder/subfolder/subfile.rw

Creating UID-Basefolder

↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder

Creating file in UID-Basefolder: ../uid_test_file

Creating folder in UID-Basefolder: ../uid_test_folder

Info Executing dirlist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

DIRLIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder)

↪ exists - looking for directories to append

DIRLIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/wr_

↪ itable_folder) exists - looking for directories to append

DIRLIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/wr_

↪ itable_folder/subfolder) exists - looking for directories to append

DIRLIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/no_

↪ t_writable_folder) exists - looking for directories to append

Success Returnvalue of dirlist is correct (Content ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_folder/subfolder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder'] and Type is <class 'list'>).

Result (Returnvalue of dirlist): ['/home/dirk/work/unittest_collection/fstools/unittest/outp_

↪ ut_data/testfolder/writable_folder',

↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_fol_

↪ der/subfolder',

↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_

↪ _folder'] (<class 'list'>)

Expectation (Returnvalue of dirlist): result = ['/home/dirk/work/unittest_collection/fstools_

↪ /unittest/output_data/testfolder/writable_folder',

↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_

↪ _folder',

↪ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_fol_

↪ der/subfolder'] (<class 'list'>)

Content '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/writable_

↪ _folder/subfolder' is incorrect for test_variable.

A.1.10 Test for fstools.dirlist (not rekursive)

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

Creating general Basefolder

↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```

Creating folder in general Basefolder: ../writable_folder
Creating folder in general Basefolder: ../not_writable_folder
Creating file in general Basefolder: ../writable_file.rw
Creating file in general Basefolder: ../not_writable_file.ro
Creating folder in general Basefolder: ../writable_folder/subfolder
Creating file in general Basefolder: ../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↳ /home/dirk/work/unitest_collection/fstools/unitest/output_data/uid_testfolder
Creating file in UID-Basefolder: ../uid_test_file
Creating folder in UID-Basefolder: ../uid_test_folder

```

Info Executing dirlist to get list for /home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder

```

DIRLIST: path (/home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder)
↳ exists - looking for directories to append

```

Success Returnvalue of dirlist is correct (Content ['/home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder/writable_folder', '/home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder/not_writable_folder'] and Type is <class 'list'>).

```

Result (Returnvalue of dirlist): [ '/home/dirk/work/unitest_collection/fstools/unitest/outp
↳ ut_data/testfolder/writable_folder',
↳ '/home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder/not_writable
↳ _folder' ] (<class 'list'>)

```

```

Expectation (Returnvalue of dirlist): result = [ '/home/dirk/work/unitest_collection/fstools
↳ /unitest/output_data/testfolder/writable_folder',
↳ '/home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder/not_writable
↳ _folder' ] (<class 'list'>)

```

A.1.11 Test for fstools.dirlist (not existing folder)

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder

```

Creating general Basefolder
↳ /home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder
Creating folder in general Basefolder: ../writable_folder
Creating folder in general Basefolder: ../not_writable_folder
Creating file in general Basefolder: ../writable_file.rw
Creating file in general Basefolder: ../not_writable_file.ro
Creating folder in general Basefolder: ../writable_folder/subfolder
Creating file in general Basefolder: ../writable_folder/subfolder/subfile.rw

```

```

Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating file in UID-Basefolder: ../uid_test_file
Creating folder in UID-Basefolder: ../uid_test_folder

```

Info Executing dirlist to get list for /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_existing_folder

DIRLIST: path (/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_existing_folder) does not exist - empty filelist will be returned

Success Returnvalue of dirlist is correct (Content [] and Type is <class 'list'>).

```

Result (Returnvalue of dirlist): [ ] (<class 'list'>)
Expectation (Returnvalue of dirlist): result = [ ] (<class 'list'>)

```

A.1.12 Test for fstools.openlocked_non_blocking

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder

```

Creating general Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder
Creating folder in general Basefolder: ../writable_folder
Creating folder in general Basefolder: ../not_writable_folder
Creating file in general Basefolder: ../writable_file.rw
Creating file in general Basefolder: ../not_writable_file.ro
Creating folder in general Basefolder: ../writable_folder/subfolder
Creating file in general Basefolder: ../writable_folder/subfolder/subfile.rw
Creating UID-Basefolder
↪ /home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder
Creating file in UID-Basefolder: ../uid_test_file
Creating folder in UID-Basefolder: ../uid_test_folder

```

Info Opening a file with a non blocking lock, writing to it, trying to open and write to it from a child process and writing more text

Success Exception name thrown in child process is correct (Content 'BlockingIOError' and Type is <class 'str'>).

```

Result (Exception name thrown in child process): 'BlockingIOError' (<class 'str'>)
Expectation (Exception name thrown in child process): result = 'BlockingIOError' (<class 'str'>)
↪ 'str'>)

```

Success Appended text to file is correct (Content 'Some more text.' and Type is <class 'str'>).

```

Result (Appended text to file): 'Some more text.' (<class 'str'>)
Expectation (Appended text to file): result = 'Some more text.' (<class 'str'>)

```

A.1.13 Test for `fstools.openlocked_blocking`

Testresult

This test was passed with the state: **Success**.

Info Files and folders for fstools unittest created under `/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder`

Creating general Basefolder

↪ `/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder`

Creating folder in general Basefolder: `.../writable_folder`

Creating folder in general Basefolder: `.../not_writable_folder`

Creating file in general Basefolder: `.../writable_file.rw`

Creating file in general Basefolder: `.../not_writable_file.ro`

Creating folder in general Basefolder: `.../writable_folder/subfolder`

Creating file in general Basefolder: `.../writable_folder/subfolder/subfile.rw`

Creating UID-Basefolder

↪ `/home/dirk/work/unittest_collection/fstools/unittest/output_data/uid_testfolder`

Creating file in UID-Basefolder: `.../uid_test_file`

Creating folder in UID-Basefolder: `.../uid_test_folder`

Info Opening a file with a blocking lock, writing to it, sleep a bit, trying to open and write to it from a child process

Success Appended text to file is correct (Content 'Some more text.' and Type is `<class 'str'>`).

Result (Appended text to file): 'Some more text.' (`<class 'str'>`)

Expectation (Appended text to file): result = 'Some more text.' (`<class 'str'>`)

B Test-Coverage

B.1 fstools

The line coverage for `fstools` was 64.4%

The branch coverage for `fstools` was 60.0%

B.1.1 `fstools.__init__.py`

The line coverage for `fstools.__init__.py` was 64.4%

The branch coverage for `fstools.__init__.py` was 60.0%

```

1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 #
4 """
5 fstools (Filesystem Tools)
6 =====
7
8 **Author:**
9
10 * Dirk Alders <sudo-dirk@mount-mockery.de>
11
12 **Description:**
13
14     This module supports functions and classes to handle files and paths
15
16 **Submodules:**
17
18 * :func:`fstools.dirlist`
19 * :func:`fstools.filelist`
20 * :func:`fstools.is_writeable`
21 * :func:`fstools.mkdir`
22 * :func:`fstools.open_locked_blocking`
23 * :func:`fstools.open_locked_non_blocking`
24 * :func:`fstools.uid`
25
26 **Unittest:**
27
28     See also the :download:`unittest <fstools/_testresults_/unittest.pdf>` documentation.
29
30 **Module Documentation:**
31
32 """
33 __DEPENDENCIES__ = []
34
35 import glob
36 import hashlib
37 import hmac
38 import logging
39 import os
40 from functools import partial
41 import sys
42 import time
43
44
45 try:
46     from config import APP_NAME as ROOT_LOGGER_NAME
47 except ImportError:
48     ROOT_LOGGER_NAME = 'root'
49 logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)
50
51
52 try:
53     import fcntl
54 except ImportError:
55     logger.warning('Importing "fcntl" was not possible. Only a limited functionality of fstools is available.')
56
57
58 __DESCRIPTION__ = """The Module {\\tt %s} is designed to help on all issues with files and folders.
59 For more Information read the documentation. """ % __name__.replace('_', '\\_')

```

```

60 """The Module Description"""
61 __INTERPRETER__ = (3, )
62 """The Tested Interpreter—Versions"""
63
64
65 def dirlist(path='.', expression='*', rekursive=True):
66     """
67     Function returning a list of directories below a given path.
68
69     :param str path: folder which is the basepath for searching files.
70     :param str expression: expression to fit including shell-style wildcards. It is only used for
71     the first directory level, if rekursive is set to True.
72     :param bool rekursive: search all subfolders if True.
73     :returns: list of filenames including the pathe
74     :rtype: list
75
76     .. note:: The returned filenames could be relative pathes depending on argument path.
77
78     **Example:**
79
80     .. literalinclude:: fstools/_examples_/dirlist.py
81
82     .. literalinclude:: fstools/_examples_/dirlist.log
83     """
84     li = list()
85     if os.path.exists(path):
86         logger.debug('DIRLIST: path (%s) exists — looking for directories to append', path)
87         for dirname in glob.glob(os.path.join(path, expression)):
88             if os.path.isdir(dirname):
89                 li.append(dirname)
90                 if rekursive:
91                     li.extend(dirlist(dirname))
92     else:
93         logger.warning('DIRLIST: path (%s) does not exist — empty filelist will be returned',
94             path)
95     return li
96
97
98 def filelist(path='.', expression='*', rekursive=True):
99     """
100     Function returning a list of files below a given path.
101
102     :param str path: folder which is the basepath for searching files.
103     :param str expression: expression to fit including shell-style wildcards.
104     :param bool rekursive: search all subfolders if True.
105     :returns: list of filenames including the pathe
106     :rtype: list
107
108     .. note:: The returned filenames could be relative pathes depending on argument path.
109
110     **Example:**
111
112     .. literalinclude:: fstools/_examples_/filelist.py
113
114     .. literalinclude:: fstools/_examples_/filelist.log
115     """
116     li = list()
117     if os.path.exists(path):
118         logger.debug('FILELIST: path (%s) exists — looking for files to append', path)
119         for filename in glob.glob(os.path.join(path, expression)):
120             if os.path.isfile(filename):

```

```

119         li.append(filename)
120     for directory in os.listdir(path):
121         directory = os.path.join(path, directory)
122         if os.path.isdir(directory) and rekursive and not os.path.islink(directory):
123             li.extend(filelist(directory, expression))
124     else:
125         logger.warning('FILELIST: path (%s) does not exist — empty filelist will be returned',
126                        path)
127     return li
128
129 def is_writeable(path):
130     """
131     Method to get the Information, if a file or folder is writable.
132
133     :param str path: file or folder to check.
134     :returns: Whether path is writable or not.
135     :rtype: bool
136
137     .. note:: If path does not exist, the return Value is :const:`False`.
138
139     **Example:**
140
141     .. literalinclude:: fstools/_examples_/is_writeable.py
142
143     .. literalinclude:: fstools/_examples_/is_writeable.log
144     """
145     if os.access(path, os.W_OK):
146         # path is writable whatever it is, file or directory
147         return True
148     else:
149         # path is not writable whatever it is, file or directory
150         return False
151
152
153 def mkdir(path):
154     """
155     Method to create a folder.
156
157     .. note:: All needed subfolders will also be created (rekursive mkdir).
158
159     :param str path: folder to be created.
160     :returns: True, if folder exists after creation commands, otherwise False.
161     :rtype: bool
162     """
163     path = os.path.abspath(path)
164     if not os.path.exists(os.path.dirname(path)):
165         mkdir(os.path.dirname(path))
166     if not os.path.exists(path):
167         os.mkdir(path)
168     return os.path.isdir(path)
169
170
171 def open_locked_blocking(*args, **kwargs):
172     """
173     Method to get exclusive access to a file.
174
175     :param args: Arguments for a standard file open call.
176     :param kwargs: Keyword arguments for a standard file open call.
177     :returns: A file descriptor.
178     :rtype: file handle
179
180     .. note:: The call blocks until file is able to be used. This can cause a deadlock, if the
181     file release es done after trying to open the file!
182     """

```

```

182     locked_file_descriptor = open(*args, **kwargs)
183    fcntl.lockf(locked_file_descriptor, fcntl.LOCK_EX)
184     return locked_file_descriptor
185
186
187 def open_locked_non_blocking(*args, **kwargs):
188     """
189     Method to get exclusive access to a file.
190
191     :param args: Arguments for a standard file open call.
192     :param kwargs: Keyword arguments for a standard file open call.
193     :raises: OSError, if the file is already blocked.
194     :returns: A file descriptor.
195     :rtype: file handle
196
197     .. note:: The call blocks until file is able to be used. This can cause a deadlock, if the
198     file release es done after trying to open the file!
199     """
200     locked_file_descriptor = open(*args, **kwargs)
201     fcntl.lockf(locked_file_descriptor, fcntl.LOCK_EX | fcntl.LOCK_NB)
202     return locked_file_descriptor
203
204 def uid(path, max_staleness=3600):
205     """
206     Function returning a "unique" id for a given file or path.
207
208     :param str path: File or folder to generate a uid for.
209     :param int max_staleness: If a file or folder is older than that, we may consider
210         it stale and return a different uid — this is a
211         dirty trick to work around changes never being
212         detected. Default is 3600 seconds, use None to
213         disable this trickery. See below for more details.
214     :returns: An object that changes value if the file changed,
215         None is returned if there were problems accessing the file
216         or folder.
217     :rtype: str
218
219     .. warning:: Depending on the operating system capabilities and the way the
220     file update is done, this function might return the same value
221     even if the file has changed. It should be better than just
222     using file's mtime though.
223     max_staleness tries to avoid the worst for these cases.
224
225     .. note:: If this function is used for a path, it will stat all pathes and files rekursively.
226
227     **Example:**
228
229     .. literalinclude:: fstools/_examples_/uid.py
230
231     .. literalinclude:: fstools/_examples_/uid.log
232
233     Using just the file's mtime to determine if the file has changed is
234     not reliable — if file updates happen faster than the file system's
235     mtime granularity, then the modification is not detectable because
236     the mtime is still the same.
237
238     This function tries to improve by using not only the mtime, but also
239     other metadata values like file size and inode to improve reliability.
240
241     For the calculation of this value, we of course only want to use data

```

```

242     that we can get rather fast, thus we use file metadata, not file data
243     (file content).
244     """
245     if os.path.isdir(path):
246         pathlist = dirlist(path) + filelist(path)
247         pathlist.sort()
248     else:
249         pathlist = [path]
250     uid = []
251     for element in pathlist:
252         try:
253             st = os.stat(element)
254         except (IOError, OSError):
255             uid.append(None) # for permanent errors on stat() this does not change, but
256                             # having a changing value would be pointless because if we
257                             # can't even stat the file, it is unlikely we can read it.
258         else:
259             fake_mtime = int(st.st_mtime)
260             if not st.st_ino and max_staleness:
261                 # st_ino being 0 likely means that we run on a platform not
262                 # supporting it (e.g. win32) — thus we likely need this dirty
263                 # trick
264                 now = int(time.time())
265                 if now >= st.st_mtime + max_staleness:
266                     # keep same fake mtime for each max staleness interval
267                     fake_mtime = int(now / max_staleness) * max_staleness
268             uid.append((
269                 st.st_mtime, # might have a rather rough granularity, e.g. 2s
270                             # on FAT, 1s on ext3 and might not change on fast
271                             # updates
272                 st.st_ino, # inode number (will change if the update is done
273                             # by e.g. renaming a temp file to the real file).
274                             # not supported on win32 (0 ever)
275                 st.st_size, # likely to change on many updates, but not
276                             # sufficient alone
277                 fake_mtime) # trick to workaround file system / platform
278                             # limitations causing permanent trouble
279             )
280     if sys.version_info < (3, 0):
281         secret = ''
282         return hmac.new(secret, repr(uid), hashlib.sha1).hexdigest()
283     else:
284         secret = b''
285         return hmac.new(secret, bytes(repr(uid), 'latin-1'), hashlib.sha1).hexdigest()
286
287
288 def uid_filelist(path='.', expression='*', rekursive=True):
289     """
290     Function returning a unique id for a given file or path.
291
292     :param str path: folder which is the basepath for searching files.
293     :param str expression: expression to fit including shell-style wildcards.
294     :param bool rekursive: search all subfolders if True.
295     :returns: An object that changes value if one of the files change.
296     :rtype: str
297
298     .. note:: This UID is created out of the file content. Therefore it is more
299               reliable than :func:`fstools.uid`, but also much slower.
300
301     **Example:**
302
303     .. literalinclude:: fstools/_examples_/uid_filelist.py
304
305     .. literalinclude:: fstools/_examples_/uid_filelist.log
306     """

```

```
307 SHAhash = hashlib.md5()
308 #
309 fl = filelist(path, expression, rekursive)
310 fl.sort()
311 for f in fl:
312     if sys.version_info < (3, 0):
313         with open(f, 'rb') as fh:
314             SHAhash.update(hashlib.md5(fh.read()).hexdigest())
315     else:
316         with open(f, mode='rb') as fh:
317             d = hashlib.md5()
318             for buf in iter(partial(fh.read, 128), b''):
319                 d.update(buf)
320             SHAhash.update(d.hexdigest().encode())
321 #
322 return SHAhash.hexdigest()
```