

Unittest for fstools

August 17, 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)	9
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)	17
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
Version	d042900afab24e63d81de00037dd783f
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	erle
Kernel	6.15.1-surface-2 (#2 SMP PREEMPT_DYNAMIC Tue Jun 24 21:02:07 UTC 2025)
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	0.524s

2.2 Coverage Statistic

Module- or Filename	Line-Coverage	Branch-Coverage
fstools	67.4%	66.7%
fstools.__init__.py	67.4%	

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 (329)
Start-Time:	2025-08-17 14:43:10,342
Finished-Time:	2025-08-17 14:43:10,345
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 .../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 (329)
Start-Time:	2025-08-17 14:43:10,345
Finished-Time:	2025-08-17 14:43:10,346
Time-Consumption	0.001s
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/unittest_collection/fstools/unittest/src/report/__init__.py (329)
Start-Time:	2025-08-17 14:43:10,357
Finished-Time:	2025-08-17 14:43:10,358
Time-Consumption	0.001s
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/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/unittest_collection/fstools/unittest/src/report/__init__.py (329)
Start-Time:	2025-08-17 14:43:10,356
Finished-Time:	2025-08-17 14:43:10,357
Time-Consumption	0.001s
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/not_writable_folder', '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/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/unittest_collection/fstools/unittest/src/report/__init__.py (329)
Start-Time:	2025-08-17 14:43:10,355
Finished-Time:	2025-08-17 14:43:10,356

Time-Consumption 0.001s

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/not_writable_folder', '/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'] 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 (329)
Start-Time:	2025-08-17 14:43:10,354
Finished-Time:	2025-08-17 14:43:10,354
Time-Consumption	0.001s

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 (329)
Start-Time:	2025-08-17 14:43:10,351
Finished-Time:	2025-08-17 14:43:10,352
Time-Consumption	0.001s

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/unitest_collection/fstools/unitest/output_data/testfolder/writable_file.rw', '/home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder/not_writable_file.ro'] 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/unitest_collection/fstools/unitest/src/report/__init__.py (329)
Start-Time:	2025-08-17 14:43:10,352
Finished-Time:	2025-08-17 14:43:10,353
Time-Consumption	0.001s
Testsummary:	
Info	Files and folders for fstools unittest created under /home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder
Info	Executing filelist to get list for /home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder
Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unitest_collection/fstools/unitest/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/unitest_collection/fstools/unitest/src/report/__init__.py (329)
Start-Time:	2025-08-17 14:43:10,347
Finished-Time:	2025-08-17 14:43:10,349
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 filelist to get list for /home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder
Success	Returnvalue of filelist is correct (Content ['/home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder/writable_file.rw', '/home/dirk/work/unitest_collection/fstools/unitest/output_data/testfolder/not_writable_file.ro', '/home/dirk/work/unitest_collection/fstools/unitest/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 (329)
Start-Time:	2025-08-17 14:43:10,349
Finished-Time:	2025-08-17 14:43:10,351
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'] 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 (329)
Start-Time:	2025-08-17 14:43:10,346
Finished-Time:	2025-08-17 14:43:10,347
Time-Consumption	0.001s
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 (329)
 Start-Time: 2025-08-17 14:43:10,362
 Finished-Time: 2025-08-17 14:43:10,867
 Time-Consumption 0.505s

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 (329)
 Start-Time: 2025-08-17 14:43:10,358
 Finished-Time: 2025-08-17 14:43:10,362
 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 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/n_`)

↪ `ot_writable_folder`) 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/writable_folder/subfolder) 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', '/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'] and Type is <class 'list'>).

```
Result (Returnvalue of filelist): [ '/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'>)
```

```
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'>)
```

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/n_
↳ ot_writable_folder) 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
```

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/out_
↳ put_data/testfolder/not_writable_file.ro' ] (<class 'list'>)
Expectation (Returnvalue of filelist): result = [ '/home/dirk/work/unittest_collection/fstool_
↳ s/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/writable_file.rw', '/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/out_
↳ put_data/testfolder/writable_file.rw',
↳ '/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/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'>)
```

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/not_writable_folder) exists - looking for directories to append
```

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

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

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

```
Result (Returnvalue of dirlist): [ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder',
↳ '/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' ] (<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_folder/subfolder' ] (<class 'list'>)
```

```
Content '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder' 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/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

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

Result (Returnvalue of dirlist): ['/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/not_writable_folder',
 ↳ '/home/dirk/work/unittest_collection/fstools/unittest/output_data/testfolder/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'] (<class 'list'>)

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

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/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/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 67.4%

The branch coverage for `fstools` was 66.7%

B.1.1 `fstools.__init__.py`

The line coverage for `fstools.__init__.py` was 67.4%

The branch coverage for `fstools.__init__.py` was 66.7%

Unittest for fstools

```
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 time
42
43
44 try:
45     from config import APP_NAME as ROOT_LOGGER_NAME
46 except ImportError:
47     ROOT_LOGGER_NAME = 'root'
48 logger = logging.getLogger(ROOT_LOGGER_NAME).getChild(__name__)
49
50
51 try:
52     import fcntl
53 except ImportError:
54     logger.warning('Importing "fcntl" was not possible. Only a limited functionality of fstools is available.')
55
56
57 __DESCRIPTION__ = """The Module {\\tt %s} is designed to help on all issues with files and folders.
58
59 For more Information read the documentation.""" % __name__.replace('_', '\\_')
60 """The Module Description"""
```

```

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

```

```

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

```

```

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

```

```

242     if os.path.isdir(path):
243         pathlist = dirlist(path) + filelist(path)
244         pathlist.sort()
245     else:
246         pathlist = [path]
247     uid = []
248     for element in pathlist:
249         try:
250             st = os.stat(element)
251         except (IOError, OSError):
252             uid.append(None)    # for permanent errors on stat() this does not change, but
253                                # having a changing value would be pointless because if we
254                                # can't even stat the file, it is unlikely we can read it.
255         else:
256             fake_mtime = int(st.st_mtime)
257             if not st.st_ino and max_staleness:
258                 # st_ino being 0 likely means that we run on a platform not
259                 # supporting it (e.g. win32) — thus we likely need this dirty
260                 # trick
261                 now = int(time.time())
262                 if now >= st.st_mtime + max_staleness:
263                     # keep same fake_mtime for each max_staleness interval
264                     fake_mtime = int(now / max_staleness) * max_staleness
265             uid.append((
266                 st.st_mtime,    # might have a rather rough granularity, e.g. 2s
267                                # on FAT, 1s on ext3 and might not change on fast
268                                # updates
269                 st.st_ino,     # inode number (will change if the update is done
270                                # by e.g. renaming a temp file to the real file).
271                                # not supported on win32 (0 ever)
272                 st.st_size,    # likely to change on many updates, but not
273                                # sufficient alone
274                 fake_mtime)    # trick to workaround file system / platform
275                                # limitations causing permanent trouble
276            )
277     secret = b''
278     return hmac.new(secret, bytes(repr(uid), 'latin-1'), hashlib.sha1).hexdigest()
279
280
281 def uid_filelist(path='.', expression='*', rekursive=True):
282     """
283     Function returning a unique id for a given file or path.
284
285     :param str path: folder which is the basepath for searching files.
286     :param str expression: expression to fit including shell-style wildcards.
287     :param bool rekursive: search all subfolders if True.
288     :returns: An object that changes value if one of the files change.
289     :rtype: str
290
291     .. note:: This UID is created out of the file content. Therefore it is more
292               reliable then :func:`fstools.uid`, but also much slower.
293
294     **Example:**
295
296     .. literalinclude:: fstools/_examples_/uid_filelist.py
297
298     .. literalinclude:: fstools/_examples_/uid_filelist.log
299     """
300     SHAhash = hashlib.md5()
301     #

```

```
302     fl = filelist(path, expression, rekursive)
303     fl.sort()
304     for f in fl:
305         with open(f, mode='rb') as fh:
306             d = hashlib.md5()
307             for buf in iter(partial(fh.read, 128), b''):
308                 d.update(buf)
309             SHAhash.update(d.hexdigest().encode())
310     #
311     return SHAhash.hexdigest()
```