:mod:`os` --- Miscellaneous operating system interfaces

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4)

Unknown directive type "module".

```
.. module:: os
   :synopsis: Miscellaneous operating system interfaces.
```

Source code: :source:`Lib/os.py`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 7); *backlink*

Unknown interpreted text role "source".

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see :func:`open`, if you want to manipulate paths, see the :mod:`os.path` module, and if you want to read all the lines in all the files on the command line see the :mod:`fileinput` module. For creating temporary files and directories see the :mod:`tempfile` module, and for high-level file and directory handling see the :mod:`shutil` module.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 11); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 11); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 11); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 11); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 11); *backlink*

Unknown interpreted text role "mod".

Notes on the availability of these functions:

- The design of all built-in operating system dependent modules of Python is such that as long as the same functionality is available, it uses the same interface; for example, the function `os.stat(path)` returns stat information about *path* in the same format (which happens to have originated with the POSIX interface).
- Extensions peculiar to a particular operating system are also available through the :mod:`os` module, but using them is of course a threat to portability.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 27); *backlink*

  Unknown interpreted text role "mod".
- All functions accepting path or file names accept both bytes and string objects, and result in an object of the same type, if a path or file name is returned.

- On VxWorks, os.popen, os.fork, os.execv and os.spawn*p* are not supported.

Note

All functions in this module raise :exc:`OSError` (or subclasses thereof) in the case of invalid or inaccessible file names and paths, or other arguments that have the correct type, but are not accepted by the operating system.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 39); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 43)

Unknown directive type "exception".

```
.. exception:: error

   An alias for the built-in :exc:`OSError` exception.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 48)

Unknown directive type "data".

```
.. data:: name

   The name of the operating system dependent module imported.  The following
   names have currently been registered: ``'posix'``, ``'nt'``,
   ``'java'``.

   .. seealso::
      :attr:`sys.platform` has a finer granularity.  :func:`os.uname` gives
      system-dependent version information.

      The :mod:`platform` module provides detailed checks for the
      system's identity.
```

File Names, Command Line Arguments, and Environment Variables

In Python, file names, command line arguments, and environment variables are represented using the string type. On some systems, decoding these strings to and from bytes is necessary before passing them to the operating system. Python uses the :term:`filesystem encoding and error handler` to perform this conversion (see :func:`sys.getfilesystemencoding`).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 68); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 68); *backlink*

Unknown interpreted text role "func".

The :term:`filesystem encoding and error handler` are configured at Python startup by the :c:func:`PyConfig_Read` function: see :c:member:`~PyConfig.filesystem_encoding` and :c:member:`~PyConfig.filesystem_errors` members of :c:type:`PyConfig`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 74); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 74); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-`

main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 74); *backlink*

Unknown interpreted text role "c:member".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 74); ***backlink***

Unknown interpreted text role "c:member".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 74); ***backlink***

Unknown interpreted text role "c:type".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 79)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.1
   On some systems, conversion using the file system encoding may fail. In this
   case, Python uses the :ref:`surrogateescape encoding error handler
   <surrogateescape>`, which means that undecodable bytes are replaced by a
   Unicode character U+DCxx on decoding, and these are again translated to the
   original byte on encoding.
```

The :term:`file system encoding <filesystem encoding and error handler>` must guarantee to successfully decode all bytes below 128. If the file system encoding fails to provide this guarantee, API functions can raise :exc:`UnicodeError`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 87); ***backlink***

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 87); ***backlink***

Unknown interpreted text role "exc".

See also the :term:`locale encoding`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 92); ***backlink***

Unknown interpreted text role "term".

Python UTF-8 Mode

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 100)

Unknown directive type "versionadded".

```
.. versionadded:: 3.7
   See :pep:`540` for more details.
```

The Python UTF-8 Mode ignores the :term:`locale encoding` and forces the usage of the UTF-8 encoding:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 103); ***backlink***

Unknown interpreted text role "term".

- Use UTF-8 as the :term:`filesystem encoding <filesystem encoding and error handler>`.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 106); ***backlink***

  Unknown interpreted text role "term".

- :func:`sys.getfilesystemencoding()` returns `'UTF-8'`.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 108); *backlink*

  Unknown interpreted text role "func".

- :func:`locale.getpreferredencoding()` returns `'UTF-8'` (the *do_setlocale* argument has no effect).

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 109); *backlink*

  Unknown interpreted text role "func".

- :data:`sys.stdin`, :data:`sys.stdout`, and :data:`sys.stderr` all use UTF-8 as their text encoding, with the `surrogateescape` :ref:`error handler <error-handlers>` being enabled for :data:`sys.stdin` and :data:`sys.stdout` (:data:`sys.stderr` continues to use `backslashreplace` as it does in the default locale-aware mode)

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 111); *backlink*

  Unknown interpreted text role "data".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 111); *backlink*

  Unknown interpreted text role "data".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 111); *backlink*

  Unknown interpreted text role "data".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 111); *backlink*

  Unknown interpreted text role "ref".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 111); *backlink*

  Unknown interpreted text role "data".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 111); *backlink*

  Unknown interpreted text role "data".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 111); *backlink*

  Unknown interpreted text role "data".

- On Unix, :func:`os.device_encoding` returns `'UTF-8'` rather than the device encoding.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 116); *backlink*

  Unknown interpreted text role "func".

Note that the standard stream settings in UTF-8 mode can be overridden by :envvar:`PYTHONIOENCODING` (just as they can be in the default locale-aware mode).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 119); ***backlink***

Unknown interpreted text role "envvar".

As a consequence of the changes in those lower level APIs, other higher level APIs also exhibit different default behaviours:

- Command line arguments, environment variables and filenames are decoded to text using the UTF-8 encoding.
- :func:`os.fsdecode()` and :func:`os.fsencode()` use the UTF-8 encoding.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 128); ***backlink***

  Unknown interpreted text role "func".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 128); ***backlink***

  Unknown interpreted text role "func".

- :func:`open()`, :func:`io.open()`, and :func:`codecs.open()` use the UTF-8 encoding by default. However, they still use the strict error handler by default so that attempting to open a binary file in text mode is likely to raise an exception rather than producing nonsense data.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 129); ***backlink***

  Unknown interpreted text role "func".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 129); ***backlink***

  Unknown interpreted text role "func".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 129); ***backlink***

  Unknown interpreted text role "func".

The :ref:`Python UTF-8 Mode <utf8-mode>` is enabled if the LC_CTYPE locale is `C` or `POSIX` at Python startup (see the :c:func:`PyConfig_Read` function).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 134); ***backlink***

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 134); ***backlink***

Unknown interpreted text role "c:func".

It can be enabled or disabled using the :option:`-X utf8 <-X>` command line option and the :envvar:`PYTHONUTF8` environment variable.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 138); ***backlink***

Unknown interpreted text role "option".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 138); ***backlink***

Unknown interpreted text role "envvar".

If the :envvar:`PYTHONUTF8` environment variable is not set at all, then the interpreter defaults to using the current locale settings, *unless* the current locale is identified as a legacy ASCII-based locale (as described for :envvar:`PYTHONCOERCECLOCALE`), and locale coercion is either disabled or fails. In such legacy locales, the interpreter will default to enabling UTF-8 mode unless explicitly instructed not to do so.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 141); ***backlink***

Unknown interpreted text role "envvar".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 141); ***backlink***

Unknown interpreted text role "envvar".

The Python UTF-8 Mode can only be enabled at the Python startup. Its value can be read from :data:`sys.flags.utf8_mode <sys.flags>`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 148); ***backlink***

Unknown interpreted text role "data".

See also the :ref:`UTF-8 mode on Windows <win-utf8-mode>` and the :term:`filesystem encoding and error handler`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 151); ***backlink***

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 151); ***backlink***

Unknown interpreted text role "term".

Process Parameters

These functions and data items provide information and operate on the current process and user.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 164)

Unknown directive type "function".

```
.. function:: ctermid()

   Return the filename corresponding to the controlling terminal of the process.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 171)

Unknown directive type "data".

```
.. data:: environ

   A :term:`mapping` object where keys and values are strings that represent
   the process environment.  For example, ``environ['HOME']`` is the pathname
   of your home directory (on some platforms), and is equivalent to
   ``getenv("HOME")`` in C.

   This mapping is captured the first time the :mod:`os` module is imported,
   typically during Python startup as part of processing :file:`site.py`.  Changes
   to the environment made after this time are not reflected in ``os.environ``,
   except for changes made by modifying ``os.environ`` directly.

   This mapping may be used to modify the environment as well as query the
   environment.  :func:`putenv` will be called automatically when the mapping
   is modified.

   On Unix, keys and values use :func:`sys.getfilesystemencoding` and
   ``'surrogateescape'`` error handler. Use :data:`environb` if you would like
   to use a different encoding.
   .. note::

      Calling :func:`putenv` directly does not change ``os.environ``, so it's better
      to modify ``os.environ``.

   .. note::

      On some platforms, including FreeBSD and macOS, setting ``environ`` may
      cause memory leaks.  Refer to the system documentation for
      :c:func:`putenv`.

   You can delete items in this mapping to unset environment variables.
   :func:`unsetenv` will be called automatically when an item is deleted from
   ``os.environ``, and when one of the :meth:`pop` or :meth:`clear` methods is
   called.

   .. versionchanged:: 3.9
      Updated to support :pep:`584`'s merge (``|``) and update (``|=``) operators.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 211)

Unknown directive type "data".

```
.. data:: environb

   Bytes version of :data:`environ`: a :term:`mapping` object where both keys
   and values are :class:`bytes` objects representing the process environment.
   :data:`environ` and :data:`environb` are synchronized (modifying
   :data:`environb` updates :data:`environ`, and vice versa).

   :data:`environb` is only available if :data:`supports_bytes_environ` is
   ``True``.

   .. versionadded:: 3.2

   .. versionchanged:: 3.9
      Updated to support :pep:`584`'s merge (``|``) and update (``|=``) operators.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 227)

Unknown directive type "function".

```
.. function:: chdir(path)
              fchdir(fd)
              getcwd()
   :noindex:

   These functions are described in :ref:`os-file-dir`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 235)

Unknown directive type "function".

```
.. function:: fsencode(filename)

   Encode :term:`path-like <path-like object>` *filename* to the
   :term:`filesystem encoding and error handler`; return :class:`bytes`
   unchanged.

   :func:`fsdecode` is the reverse function.

   .. versionadded:: 3.2

   .. versionchanged:: 3.6
      Support added to accept objects implementing the :class:`os.PathLike`
      interface.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 250)

Unknown directive type "function".

```
.. function:: fsdecode(filename)
```

```
   Decode the :term:`path-like <path-like object>` *filename* from the
   :term:`filesystem encoding and error handler`; return :class:`str`
   unchanged.

   :func:`fsencode` is the reverse function.

   .. versionadded:: 3.2

   .. versionchanged:: 3.6
      Support added to accept objects implementing the :class:`os.PathLike`
      interface.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 265)

Unknown directive type "function".

```
.. function:: fspath(path)

   Return the file system representation of the path.

   If :class:`str` or :class:`bytes` is passed in, it is returned unchanged.
   Otherwise :meth:`~os.PathLike.__fspath__` is called and its value is
   returned as long as it is a :class:`str` or :class:`bytes` object.
   In all other cases, :exc:`TypeError` is raised.

   .. versionadded:: 3.6
```

An :term:`abstract base class` for objects representing a file system path, e.g. :class:`pathlib.PurePath`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 279); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 279); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 282)

Unknown directive type "versionadded".

```
.. versionadded:: 3.6
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 284)

Unknown directive type "abstractmethod".

```
.. abstractmethod:: __fspath__()

   Return the file system path representation of the object.

   The method should only return a :class:`str` or :class:`bytes` object,
   with the preference being for :class:`str`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 292)

Unknown directive type "function".

```
.. function:: getenv(key, default=None)

   Return the value of the environment variable *key* if it exists, or
   *default* if it doesn't. *key*, *default* and the result are str.

   On Unix, keys and values are decoded with :func:`sys.getfilesystemencoding`
   and ``'surrogateescape'`` error handler. Use :func:`os.getenvb` if you
   would like to use a different encoding.

   .. availability:: most flavors of Unix, Windows.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 304)

Unknown directive type "function".

```
.. function:: getenvb(key, default=None)

   Return the value of the environment variable *key* if it exists, or
   *default* if it doesn't. *key*, *default* and the result are bytes.

   :func:`getenvb` is only available if :data:`supports_bytes_environ`
   is ``True``.

   .. availability:: most flavors of Unix.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 317)

Unknown directive type "function".

```
.. function:: get_exec_path(env=None)

   Returns the list of directories that will be searched for a named
   executable, similar to a shell, when launching a process.
   *env*, when specified, should be an environment variable dictionary
   to lookup the PATH in.
   By default, when *env* is ``None``, :data:`environ` is used.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 328)

Unknown directive type "function".

```
.. function:: getegid()

   Return the effective group id of the current process.  This corresponds to the
   "set id" bit on the file being executed in the current process.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 336)

Unknown directive type "function".

```
.. function:: geteuid()

   .. index:: single: user; effective id

   Return the current process's effective user id.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 345)

Unknown directive type "function".

```
.. function:: getgid()

   .. index:: single: process; group

   Return the real group id of the current process.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 354)

Unknown directive type "function".

```
.. function:: getgrouplist(user, group)

   Return list of group ids that *user* belongs to. If *group* is not in the
   list, it is included; typically, *group* is specified as the group ID
   field from the password record for *user*.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 365)

Unknown directive type "function".

```
.. function:: getgroups()

   Return list of supplemental group ids associated with the current process.

   .. availability:: Unix.

   .. note::

      On macOS, :func:`getgroups` behavior differs somewhat from
      other Unix platforms. If the Python interpreter was built with a
      deployment target of :const:`10.5` or earlier, :func:`getgroups` returns
      the list of effective group ids associated with the current user process;
      this list is limited to a system-defined number of entries, typically 16,
      and may be modified by calls to :func:`setgroups` if suitably privileged.
      If built with a deployment target greater than :const:`10.5`,
      :func:`getgroups` returns the current group access list for the user
      associated with the effective user id of the process; the group access
      list may change over the lifetime of the process, it is not affected by
      calls to :func:`setgroups`, and its length is not limited to 16.  The
      deployment target value, :const:`MACOSX_DEPLOYMENT_TARGET`, can be
      obtained with :func:`sysconfig.get_config_var`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 388)

Unknown directive type "function".

```
.. function:: getlogin()

   Return the name of the user logged in on the controlling terminal of the
   process.  For most purposes, it is more useful to use
   :func:`getpass.getuser` since the latter checks the environment variables
   :envvar:`LOGNAME` or :envvar:`USERNAME` to find out who the user is, and
   falls back to ``pwd.getpwuid(os.getuid())[0]`` to get the login name of the
   current real user id.

   .. availability:: Unix, Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 400)

Unknown directive type "function".

```
.. function:: getpgid(pid)

   Return the process group id of the process with process id *pid*. If *pid* is 0,
   the process group id of the current process is returned.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 407)

Unknown directive type "function".

```
.. function:: getpgrp()

   .. index:: single: process; group

   Return the id of the current process group.
   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 416)

Unknown directive type "function".

```
.. function:: getpid()

   .. index:: single: process; id

   Return the current process id.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 423)

Unknown directive type "function".

```
.. function:: getppid()

   .. index:: single: process; id of parent

   Return the parent's process id.  When the parent process has exited, on Unix
   the id returned is the one of the init process (1), on Windows it is still
   the same id, which may be already reused by another process.

   .. availability:: Unix, Windows.

   .. versionchanged:: 3.2
      Added support for Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 437)

Unknown directive type "function".

```
.. function:: getpriority(which, who)

   .. index:: single: process; scheduling priority

   Get program scheduling priority.  The value *which* is one of
   :const:`PRIO_PROCESS`, :const:`PRIO_PGRP`, or :const:`PRIO_USER`, and *who*
   is interpreted relative to *which* (a process identifier for
   :const:`PRIO_PROCESS`, process group identifier for :const:`PRIO_PGRP`, and a
   user ID for :const:`PRIO_USER`).  A zero value for *who* denotes
   (respectively) the calling process, the process group of the calling process,
   or the real user ID of the calling process.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 454)

Unknown directive type "data".

```
.. data:: PRIO_PROCESS
          PRIO_PGRP
          PRIO_USER

   Parameters for the :func:`getpriority` and :func:`setpriority` functions.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 465)

Unknown directive type "function".

```
.. function:: getresuid()
```

```
   Return a tuple (ruid, euid, suid) denoting the current process's
   real, effective, and saved user ids.

   .. availability:: Unix.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 475)

Unknown directive type "function".

```
.. function:: getresgid()

   Return a tuple (rgid, egid, sgid) denoting the current process's
   real, effective, and saved group ids.

   .. availability:: Unix.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 485)

Unknown directive type "function".

```
.. function:: getuid()

   .. index:: single: user; id

   Return the current process's real user id.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 494)

Unknown directive type "function".

```
.. function:: initgroups(username, gid)

   Call the system initgroups() to initialize the group access list with all of
   the groups of which the specified username is a member, plus the specified
   group id.

   .. availability:: Unix.

   .. versionadded:: 3.2
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 505)

Unknown directive type "function".

```
.. function:: putenv(key, value)

   .. index:: single: environment variables; setting

   Set the environment variable named *key* to the string *value*.  Such
   changes to the environment affect subprocesses started with :func:`os.system`,
   :func:`popen` or :func:`fork` and :func:`execv`.

   Assignments to items in ``os.environ`` are automatically translated into
   corresponding calls to :func:`putenv`; however, calls to :func:`putenv`
   don't update ``os.environ``, so it is actually preferable to assign to items
   of ``os.environ``.

   .. note::

      On some platforms, including FreeBSD and macOS, setting ``environ`` may
      cause memory leaks. Refer to the system documentation for :c:func:`putenv`.

   .. audit-event:: os.putenv key,value os.putenv

   .. versionchanged:: 3.9
      The function is now always available.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 529)

Unknown directive type "function".

```
.. function:: setegid(egid)

   Set the current process's effective group id.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 536)

Unknown directive type "function".

```
.. function:: seteuid(euid)

   Set the current process's effective user id.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 543)

Unknown directive type "function".

```
.. function:: setgid(gid)

   Set the current process' group id.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 550)

Unknown directive type "function".

```
.. function:: setgroups(groups)

   Set the list of supplemental group ids associated with the current process to
   *groups*. *groups* must be a sequence, and each element must be an integer
   identifying a group. This operation is typically available only to the superuser.

   .. availability:: Unix.

   .. note:: On macOS, the length of *groups* may not exceed the
      system-defined maximum number of effective group ids, typically 16.
      See the documentation for :func:`getgroups` for cases where it may not
      return the same group list set by calling setgroups().
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 563)

Unknown directive type "function".

```
.. function:: setpgrp()

   Call the system call :c:func:`setpgrp` or ``setpgrp(0, 0)`` depending on
   which version is implemented (if any).  See the Unix manual for the semantics.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 571)

Unknown directive type "function".

```
.. function:: setpgid(pid, pgrp)

   Call the system call :c:func:`setpgid` to set the process group id of the
   process with id *pid* to the process group with id *pgrp*.  See the Unix manual
```

```
      for the semantics.

      .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 580)

Unknown directive type "function".

```
   .. function:: setpriority(which, who, priority)

      .. index:: single: process; scheduling priority

      Set program scheduling priority. The value *which* is one of
      :const:`PRIO_PROCESS`, :const:`PRIO_PGRP`, or :const:`PRIO_USER`, and *who*
      is interpreted relative to *which* (a process identifier for
      :const:`PRIO_PROCESS`, process group identifier for :const:`PRIO_PGRP`, and a
      user ID for :const:`PRIO_USER`). A zero value for *who* denotes
      (respectively) the calling process, the process group of the calling process,
      or the real user ID of the calling process.
      *priority* is a value in the range -20 to 19. The default priority is 0;
      lower priorities cause more favorable scheduling.

      .. availability:: Unix.

      .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 599)

Unknown directive type "function".

```
   .. function:: setregid(rgid, egid)

      Set the current process's real and effective group ids.

      .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 606)

Unknown directive type "function".

```
   .. function:: setresgid(rgid, egid, sgid)

      Set the current process's real, effective, and saved group ids.

      .. availability:: Unix.

      .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 615)

Unknown directive type "function".

```
   .. function:: setresuid(ruid, euid, suid)

      Set the current process's real, effective, and saved user ids.

      .. availability:: Unix.

      .. versionadded:: 3.2
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 624)

Unknown directive type "function".

```
   .. function:: setreuid(ruid, euid)

      Set the current process's real and effective user ids.

      .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 631)

Unknown directive type "function".

```
.. function:: getsid(pid)

   Call the system call :c:func:`getsid`.  See the Unix manual for the semantics.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 638)

Unknown directive type "function".

```
.. function:: setsid()

   Call the system call :c:func:`setsid`.  See the Unix manual for the semantics.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 645)

Unknown directive type "function".

```
.. function:: setuid(uid)

   .. index:: single: user; id, setting

   Set the current process's user id.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 655)

Unknown directive type "function".

```
.. function:: strerror(code)

   Return the error message corresponding to the error code in *code*.
   On platforms where :c:func:`strerror` returns ``NULL`` when given an unknown
   error number, :exc:`ValueError` is raised.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 662)

Unknown directive type "data".

```
.. data:: supports_bytes_environ

   ``True`` if the native OS type of the environment is bytes (eg. ``False`` on
   Windows).

   .. versionadded:: 3.2
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 670)

Unknown directive type "function".

```
.. function:: umask(mask)

   Set the current numeric umask and return the previous umask.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 675)

Unknown directive type "function".

```
.. function:: uname()

   .. index::
      single: gethostname() (in module socket)
      single: gethostbyaddr() (in module socket)

   Returns information identifying the current operating system.
   The return value is an object with five attributes:

   * :attr:`sysname` - operating system name
   * :attr:`nodename` - name of machine on network (implementation-defined)
   * :attr:`release` - operating system release
   * :attr:`version` - operating system version
   * :attr:`machine` - hardware identifier

   For backwards compatibility, this object is also iterable, behaving
   like a five-tuple containing :attr:`sysname`, :attr:`nodename`,
   :attr:`release`, :attr:`version`, and :attr:`machine`
   in that order.

   Some systems truncate :attr:`nodename` to 8 characters or to the
   leading component; a better way to get the hostname is
   :func:`socket.gethostname`  or even
   ``socket.gethostbyaddr(socket.gethostname())``.

   .. availability:: recent flavors of Unix.

   .. versionchanged:: 3.3
      Return type changed from a tuple to a tuple-like object
      with named attributes.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 707)

Unknown directive type "function".

```
.. function:: unsetenv(key)

   .. index:: single: environment variables; deleting

   Unset (delete) the environment variable named *key*. Such changes to the
   environment affect subprocesses started with :func:`os.system`, :func:`popen` or
   :func:`fork` and :func:`execv`.

   Deletion of items in ``os.environ`` is automatically translated into a
   corresponding call to :func:`unsetenv`; however, calls to :func:`unsetenv`
   don't update ``os.environ``, so it is actually preferable to delete items of
   ``os.environ``.

   .. audit-event:: os.unsetenv key os.unsetenv

   .. versionchanged:: 3.9
      The function is now always available and is also available on Windows.
```

File Object Creation

These functions create new :term:`file objects <file object>`. (See also :func:`~os.open` for opening file descriptors.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 731); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 731); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 735)

Unknown directive type "function".

```
.. function:: fdopen(fd, *args, **kwargs)

   Return an open file object connected to the file descriptor *fd*.  This is an
   alias of the :func:`open` built-in function and accepts the same arguments.
   The only difference is that the first argument of :func:`fdopen` must always
   be an integer.
```

File Descriptor Operations

These functions operate on I/O streams referenced using file descriptors.

File descriptors are small integers corresponding to a file that has been opened by the current process. For example, standard input is usually file descriptor 0, standard output is 1, and standard error is 2. Further files opened by a process will then be assigned 3, 4, 5, and so forth. The name "file descriptor" is slightly deceptive; on Unix platforms, sockets and pipes are also referenced by file descriptors.

The :meth:`~io.IOBase.fileno` method can be used to obtain the file descriptor associated with a :term:`file object` when required. Note that using the file descriptor directly will bypass the file object methods, ignoring aspects such as internal buffering of data.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 757); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 757); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 763)

Unknown directive type "function".

```
.. function:: close(fd)

   Close file descriptor *fd*.

   .. note::

      This function is intended for low-level I/O and must be applied to a file
      descriptor as returned by :func:`os.open` or :func:`pipe`.  To close a "file
      object" returned by the built-in function :func:`open` or by :func:`popen` or
      :func:`fdopen`, use its :meth:`~io.IOBase.close` method.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 775)

Unknown directive type "function".

```
.. function:: closerange(fd_low, fd_high)

   Close all file descriptors from *fd_low* (inclusive) to *fd_high* (exclusive),
   ignoring errors. Equivalent to (but much faster than)::

      for fd in range(fd_low, fd_high):
          try:
              os.close(fd)
          except OSError:
              pass
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 787)

Unknown directive type "function".

```
.. function:: copy_file_range(src, dst, count, offset_src=None, offset_dst=None)

   Copy *count* bytes from file descriptor *src*, starting from offset
   *offset_src*, to file descriptor *dst*, starting from offset *offset_dst*.
   If *offset_src* is None, then *src* is read from the current position;
   respectively for *offset_dst*. The files pointed by *src* and *dst*
   must reside in the same filesystem, otherwise an :exc:`OSError` is
   raised with :attr:`~OSError.errno` set to :data:`errno.EXDEV`.
   This copy is done without the additional cost of transferring data
   from the kernel to user space and then back into the kernel. Additionally,
   some filesystems could implement extra optimizations. The copy is done as if
   both files are opened as binary.

   The return value is the amount of bytes copied. This could be less than the
   amount requested.

   .. availability:: Linux kernel >= 4.5 or glibc >= 2.27.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 809)

Unknown directive type "function".

```
.. function:: device_encoding(fd)

   Return a string describing the encoding of the device associated with *fd*
   if it is connected to a terminal; else return :const:`None`.

   On Unix, if the :ref:`Python UTF-8 Mode <utf8-mode>` is enabled, return
   ``'UTF-8'`` rather than the device encoding.

   .. versionchanged:: 3.10
      On Unix, the function now implements the Python UTF-8 Mode.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 821)

Unknown directive type "function".

```
.. function:: dup(fd)

   Return a duplicate of file descriptor *fd*. The new file descriptor is
   :ref:`non-inheritable <fd_inheritance>`.

   On Windows, when duplicating a standard stream (0: stdin, 1: stdout,
   2: stderr), the new file descriptor is :ref:`inheritable
   <fd_inheritance>`.

   .. versionchanged:: 3.4
      The new file descriptor is now non-inheritable.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 834)

Unknown directive type "function".

```
.. function:: dup2(fd, fd2, inheritable=True)

   Duplicate file descriptor *fd* to *fd2*, closing the latter first if
   necessary. Return *fd2*. The new file descriptor is :ref:`inheritable
   <fd_inheritance>` by default or non-inheritable if *inheritable*
   is ``False``.

   .. versionchanged:: 3.4
      Add the optional *inheritable* parameter.

   .. versionchanged:: 3.7
      Return *fd2* on success. Previously, ``None`` was always returned.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 848)

Unknown directive type "function".

```
.. function:: fchmod(fd, mode)

   Change the mode of the file given by *fd* to the numeric *mode*.  See the
   docs for :func:`chmod` for possible values of *mode*.  As of Python 3.3, this
   is equivalent to ``os.chmod(fd, mode)``.

   .. audit-event:: os.chmod path,mode,dir_fd os.fchmod

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 859)

Unknown directive type "function".

```
.. function:: fchown(fd, uid, gid)

   Change the owner and group id of the file given by *fd* to the numeric *uid*
   and *gid*.  To leave one of the ids unchanged, set it to -1.  See
   :func:`chown`.  As of Python 3.3, this is equivalent to ``os.chown(fd, uid,
   gid)``.

   .. audit-event:: os.chown path,uid,gid,dir_fd os.fchown

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 871)

Unknown directive type "function".

```
.. function:: fdatasync(fd)

   Force write of file with filedescriptor *fd* to disk. Does not force update of
   metadata.

   .. availability:: Unix.

   .. note::
      This function is not available on MacOS.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 882)

Unknown directive type "function".

```
.. function:: fpathconf(fd, name)

   Return system configuration information relevant to an open file. *name*
   specifies the configuration value to retrieve; it may be a string which is the
   name of a defined system value; these names are specified in a number of
   standards (POSIX.1, Unix 95, Unix 98, and others).  Some platforms define
   additional names as well.  The names known to the host operating system are
   given in the ``pathconf_names`` dictionary.  For configuration variables not
   included in that mapping, passing an integer for *name* is also accepted.

   If *name* is a string and is not known, :exc:`ValueError` is raised.  If a
   specific value for *name* is not supported by the host system, even if it is
   included in ``pathconf_names``, an :exc:`OSError` is raised with
   :const:`errno.EINVAL` for the error number.

   As of Python 3.3, this is equivalent to ``os.pathconf(fd, name)``.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 902)

Unknown directive type "function".

```
.. function:: fstat(fd)

   Get the status of the file descriptor *fd*. Return a :class:`stat_result`
   object.

   As of Python 3.3, this is equivalent to ``os.stat(fd)``.

   .. seealso::

      The :func:`.stat` function.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 914)

Unknown directive type "function".

```
.. function:: fstatvfs(fd)

   Return information about the filesystem containing the file associated with
   file descriptor *fd*, like :func:`statvfs`.  As of Python 3.3, this is
   equivalent to ``os.statvfs(fd)``.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 923)

Unknown directive type "function".

```
.. function:: fsync(fd)

   Force write of file with filedescriptor *fd* to disk.  On Unix, this calls the
   native :c:func:`fsync` function; on Windows, the MS :c:func:`_commit` function.

   If you're starting with a buffered Python :term:`file object` *f*, first do
   ``f.flush()``, and then do ``os.fsync(f.fileno())``, to ensure that all internal
   buffers associated with *f* are written to disk.

   .. availability:: Unix, Windows.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 935)

Unknown directive type "function".

```
.. function:: ftruncate(fd, length)

   Truncate the file corresponding to file descriptor *fd*, so that it is at
   most *length* bytes in size.  As of Python 3.3, this is equivalent to
   ``os.truncate(fd, length)``.

   .. audit-event:: os.truncate fd,length os.ftruncate

   .. availability:: Unix, Windows.

   .. versionchanged:: 3.5
      Added support for Windows
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 949)

Unknown directive type "function".

```
.. function:: get_blocking(fd)

   Get the blocking mode of the file descriptor: ``False`` if the
   :data:`O_NONBLOCK` flag is set, ``True`` if the flag is cleared.

   See also :func:`set_blocking` and :meth:`socket.socket.setblocking`.

   .. availability:: Unix.

   .. versionadded:: 3.5
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 961)

Unknown directive type "function".

```
.. function:: isatty(fd)

   Return ``True`` if the file descriptor *fd* is open and connected to a
   tty(-like) device, else ``False``.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 967)

Unknown directive type "function".

```
.. function:: lockf(fd, cmd, len)

   Apply, test or remove a POSIX lock on an open file descriptor.
   *fd* is an open file descriptor.
   *cmd* specifies the command to use - one of :data:`F_LOCK`, :data:`F_TLOCK`,
   :data:`F_ULOCK` or :data:`F_TEST`.
   *len* specifies the section of the file to lock.

   .. audit-event:: os.lockf fd,cmd,len os.lockf

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 982)

Unknown directive type "data".

```
.. data:: F_LOCK
          F_TLOCK
          F_ULOCK
          F_TEST

   Flags that specify what action :func:`lockf` will take.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 994)

Unknown directive type "function".

```
.. function:: lseek(fd, pos, how)

   Set the current position of file descriptor *fd* to position *pos*, modified
   by *how*: :const:`SEEK_SET` or ``0`` to set the position relative to the
   beginning of the file; :const:`SEEK_CUR` or ``1`` to set it relative to the
   current position; :const:`SEEK_END` or ``2`` to set it relative to the end of
   the file. Return the new cursor position in bytes, starting from the beginning.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1003)

Unknown directive type "data".

```
.. data:: SEEK_SET
          SEEK_CUR
          SEEK_END

   Parameters to the :func:`lseek` function. Their values are 0, 1, and 2,
   respectively.

   .. versionadded:: 3.3
      Some operating systems could support additional values, like
      :data:`os.SEEK_HOLE` or :data:`os.SEEK_DATA`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1015)

Unknown directive type "function".

```
.. function:: open(path, flags, mode=0o777, *, dir_fd=None)

   Open the file *path* and set various flags according to *flags* and possibly
   its mode according to *mode*.  When computing *mode*, the current umask value
   is first masked out.  Return the file descriptor for the newly opened file.
   The new file descriptor is :ref:`non-inheritable <fd_inheritance>`.

   For a description of the flag and mode values, see the C run-time documentation;
   flag constants (like :const:`O_RDONLY` and :const:`O_WRONLY`) are defined in
   the :mod:`os` module.  In particular, on Windows adding
   :const:`O_BINARY` is needed to open files in binary mode.
   This function can support :ref:`paths relative to directory descriptors
   <dir_fd>` with the *dir_fd* parameter.

   .. audit-event:: open path,mode,flags os.open

   .. versionchanged:: 3.4
      The new file descriptor is now non-inheritable.

   .. note::

      This function is intended for low-level I/O.  For normal usage, use the
      built-in function :func:`open`, which returns a :term:`file object` with
      :meth:`~file.read` and :meth:`~file.write` methods (and many more).  To
      wrap a file descriptor in a file object, use :func:`fdopen`.

   .. versionadded:: 3.3
      The *dir_fd* argument.

   .. versionchanged:: 3.5
      If the system call is interrupted and the signal handler does not raise an
      exception, the function now retries the system call instead of raising an
      :exc:`InterruptedError` exception (see :pep:`475` for the rationale).

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

The following constants are options for the *flags* parameter to the :func:`~os.open` function. They can be combined using the bitwise OR operator `|`. Some of them are not available on all platforms. For descriptions of their availability and use, consult the :manpage:`open(2)` manual page on Unix or the MSDN on Windows.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1053); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1053); *backlink*

Unknown interpreted text role "manpage".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1060)

Unknown directive type "data".

```
.. data:: O_RDONLY
          O_WRONLY
          O_RDWR
          O_APPEND
          O_CREAT
          O_EXCL
          O_TRUNC

   The above constants are available on Unix and Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1071)

Unknown directive type "data".

```
.. data:: O_DSYNC
          O_RSYNC
          O_SYNC
          O_NDELAY
          O_NONBLOCK
          O_NOCTTY
          O_CLOEXEC

   The above constants are only available on Unix.

   .. versionchanged:: 3.3
      Add :data:`O_CLOEXEC` constant.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1084)

Unknown directive type "data".

```
.. data:: O_BINARY
          O_NOINHERIT
          O_SHORT_LIVED
          O_TEMPORARY
          O_RANDOM
          O_SEQUENTIAL
          O_TEXT

   The above constants are only available on Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1094)

Unknown directive type "data".

```
.. data:: O_EVTONLY
          O_FSYNC
          O_SYMLINK
          O_NOFOLLOW_ANY

   The above constants are only available on macOS.

   .. versionchanged:: 3.10
      Add :data:`O_EVTONLY`, :data:`O_FSYNC`, :data:`O_SYMLINK`
      and :data:`O_NOFOLLOW_ANY` constants.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1105)

Unknown directive type "data".

```
.. data:: O_ASYNC
          O_DIRECT
          O_DIRECTORY
          O_NOFOLLOW
          O_NOATIME
          O_PATH
          O_TMPFILE
          O_SHLOCK
          O_EXLOCK

   The above constants are extensions and not present if they are not defined by
   the C library.

   .. versionchanged:: 3.4
      Add :data:`O_PATH` on systems that support it.
      Add :data:`O_TMPFILE`, only available on Linux Kernel 3.11
        or newer.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1124)

Unknown directive type "function".

```
.. function:: openpty()

   .. index:: module: pty

   Open a new pseudo-terminal pair. Return a pair of file descriptors
   ``(master, slave)`` for the pty and the tty, respectively. The new file
   descriptors are :ref:`non-inheritable <fd_inheritance>`. For a (slightly) more
   portable approach, use the :mod:`pty` module.

   .. availability:: some flavors of Unix.

   .. versionchanged:: 3.4
      The new file descriptors are now non-inheritable.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1139)

Unknown directive type "function".

```
.. function:: pipe()

   Create a pipe.  Return a pair of file descriptors ``(r, w)`` usable for
   reading and writing, respectively. The new file descriptor is
   :ref:`non-inheritable <fd_inheritance>`.
   .. availability:: Unix, Windows.

   .. versionchanged:: 3.4
      The new file descriptors are now non-inheritable.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1151)

Unknown directive type "function".

```
.. function:: pipe2(flags)

   Create a pipe with *flags* set atomically.
   *flags* can be constructed by ORing together one or more of these values:
   :data:`O_NONBLOCK`, :data:`O_CLOEXEC`.
   Return a pair of file descriptors ``(r, w)`` usable for reading and writing,
   respectively.

   .. availability:: some flavors of Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1164)

Unknown directive type "function".

```
.. function:: posix_fallocate(fd, offset, len)

   Ensures that enough disk space is allocated for the file specified by *fd*
   starting from *offset* and continuing for *len* bytes.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1174)

Unknown directive type "function".

```
.. function:: posix_fadvise(fd, offset, len, advice)

   Announces an intention to access data in a specific pattern thus allowing
   the kernel to make optimizations.
   The advice applies to the region of the file specified by *fd* starting at
   *offset* and continuing for *len* bytes.
   *advice* is one of :data:`POSIX_FADV_NORMAL`, :data:`POSIX_FADV_SEQUENTIAL`,
   :data:`POSIX_FADV_RANDOM`, :data:`POSIX_FADV_NOREUSE`,
   :data:`POSIX_FADV_WILLNEED` or :data:`POSIX_FADV_DONTNEED`.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1189)

Unknown directive type "data".

```
.. data:: POSIX_FADV_NORMAL
          POSIX_FADV_SEQUENTIAL
          POSIX_FADV_RANDOM
          POSIX_FADV_NOREUSE
          POSIX_FADV_WILLNEED
          POSIX_FADV_DONTNEED

   Flags that can be used in *advice* in :func:`posix_fadvise` that specify
   the access pattern that is likely to be used.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1204)

Unknown directive type "function".

```
.. function:: pread(fd, n, offset)

   Read at most *n* bytes from file descriptor *fd* at a position of *offset*,
   leaving the file offset unchanged.

   Return a bytestring containing the bytes read. If the end of the file
   referred to by *fd* has been reached, an empty bytes object is returned.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1217)

Unknown directive type "function".

```
.. function:: preadv(fd, buffers, offset, flags=0)

   Read from a file descriptor *fd* at a position of *offset* into mutable
   :term:`bytes-like objects <bytes-like object>` *buffers*, leaving the file
   offset unchanged.  Transfer data into each buffer until it is full and then
   move on to the next buffer in the sequence to hold the rest of the data.

   The flags argument contains a bitwise OR of zero or more of the following
   flags:

   - :data:`RWF_HIPRI`
   - :data:`RWF_NOWAIT`

   Return the total number of bytes actually read which can be less than the
   total capacity of all the objects.

   The operating system may set a limit (:func:`sysconf` value
   ``'SC_IOV_MAX'``) on the number of buffers that can be used.

   Combine the functionality of :func:`os.readv` and :func:`os.pread`.

   .. availability:: Linux 2.6.30 and newer, FreeBSD 6.0 and newer,
      OpenBSD 2.7 and newer, AIX 7.1 and newer. Using flags requires
      Linux 4.6 or newer.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1245)

Unknown directive type "data".

```
.. data:: RWF_NOWAIT

   Do not wait for data which is not immediately available. If this flag is
   specified, the system call will return instantly if it would have to read
   data from the backing storage or wait for a lock.

   If some data was successfully read, it will return the number of bytes read.
   If no bytes were read, it will return ``-1`` and set errno to
   :data:`errno.EAGAIN`.

   .. availability:: Linux 4.14 and newer.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1260)

Unknown directive type "data".

```
.. data:: RWF_HIPRI

   High priority read/write. Allows block-based filesystems to use polling
   of the device, which provides lower latency, but may use additional
   resources.
   Currently, on Linux, this feature is usable only on a file descriptor opened
   using the :data:`O_DIRECT` flag.

   .. availability:: Linux 4.6 and newer.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1274)

Unknown directive type "function".

```
.. function:: pwrite(fd, str, offset)

   Write the bytestring in *str* to file descriptor *fd* at position of
   *offset*, leaving the file offset unchanged.

   Return the number of bytes actually written.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1286)

Unknown directive type "function".

```
.. function:: pwritev(fd, buffers, offset, flags=0)

   Write the *buffers* contents to file descriptor *fd* at a offset *offset*,
   leaving the file offset unchanged.  *buffers* must be a sequence of
   :term:`bytes-like objects <bytes-like object>`. Buffers are processed in
   array order. Entire contents of the first buffer is written before
   proceeding to the second, and so on.

   The flags argument contains a bitwise OR of zero or more of the following
   flags:

   - :data:`RWF_DSYNC`
   - :data:`RWF_SYNC`
   - :data:`RWF_APPEND`

   Return the total number of bytes actually written.

   The operating system may set a limit (:func:`sysconf` value
   ``'SC_IOV_MAX'``) on the number of buffers that can be used.

   Combine the functionality of :func:`os.writev` and :func:`os.pwrite`.

   .. availability:: Linux 2.6.30 and newer, FreeBSD 6.0 and newer,
      OpenBSD 2.7 and newer, AIX 7.1 and newer. Using flags requires
      Linux 4.7 or newer.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1315)

Unknown directive type "data".

```
.. data:: RWF_DSYNC

   Provide a per-write equivalent of the :data:`O_DSYNC` :func:`os.open` flag.
   This flag effect applies only to the data range written by the system call.

   .. availability:: Linux 4.7 and newer.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1325)

Unknown directive type "data".

```
.. data:: RWF_SYNC
```

```
   Provide a per-write equivalent of the :data:`O_SYNC` :func:`os.open` flag.
   This flag effect applies only to the data range written by the system call.

   .. availability:: Linux 4.7 and newer.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1335)

Unknown directive type "data".

```
.. data:: RWF_APPEND

   Provide a per-write equivalent of the :data:`O_APPEND` :func:`os.open`
   flag. This flag is meaningful only for :func:`os.pwritev`, and its
   effect applies only to the data range written by the system call. The
   *offset* argument does not affect the write operation; the data is always
   appended to the end of the file. However, if the *offset* argument is
   ``-1``, the current file *offset* is updated.

   .. availability:: Linux 4.16 and newer.

   .. versionadded:: 3.10
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1349)

Unknown directive type "function".

```
.. function:: read(fd, n)

   Read at most *n* bytes from file descriptor *fd*.

   Return a bytestring containing the bytes read. If the end of the file
   referred to by *fd* has been reached, an empty bytes object is returned.

   .. note::

      This function is intended for low-level I/O and must be applied to a file
      descriptor as returned by :func:`os.open` or :func:`pipe`.  To read a
      "file object" returned by the built-in function :func:`open` or by
      :func:`popen` or :func:`fdopen`, or :data:`sys.stdin`, use its
      :meth:`~file.read` or :meth:`~file.readline` methods.

   .. versionchanged:: 3.5
      If the system call is interrupted and the signal handler does not raise an
      exception, the function now retries the system call instead of raising an
      :exc:`InterruptedError` exception (see :pep:`475` for the rationale).
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1370)

Unknown directive type "function".

```
.. function:: sendfile(out_fd, in_fd, offset, count)
              sendfile(out_fd, in_fd, offset, count, headers=(), trailers=(), flags=0)

   Copy *count* bytes from file descriptor *in_fd* to file descriptor *out_fd*
   starting at *offset*.
   Return the number of bytes sent. When EOF is reached return ``0``.

   The first function notation is supported by all platforms that define
   :func:`sendfile`.

   On Linux, if *offset* is given as ``None``, the bytes are read from the
   current position of *in_fd* and the position of *in_fd* is updated.

   The second case may be used on macOS and FreeBSD where *headers* and
   *trailers* are arbitrary sequences of buffers that are written before and
   after the data from *in_fd* is written. It returns the same as the first case.

   On macOS and FreeBSD, a value of ``0`` for *count* specifies to send until
   the end of *in_fd* is reached.

   All platforms support sockets as *out_fd* file descriptor, and some platforms
   allow other types (e.g. regular file, pipe) as well.

   Cross-platform applications should not use *headers*, *trailers* and *flags*
   arguments.
   .. availability:: Unix.

   .. note::

      For a higher-level wrapper of :func:`sendfile`, see
      :meth:`socket.socket.sendfile`.

   .. versionadded:: 3.3

   .. versionchanged:: 3.9
      Parameters *out* and *in* was renamed to *out_fd* and *in_fd*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1409)

Unknown directive type "function".

```
.. function:: set_blocking(fd, blocking)

   Set the blocking mode of the specified file descriptor. Set the
   :data:`O_NONBLOCK` flag if blocking is ``False``, clear the flag otherwise.

   See also :func:`get_blocking` and :meth:`socket.socket.setblocking`.

   .. availability:: Unix.

   .. versionadded:: 3.5
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1421)

Unknown directive type "data".

```
.. data:: SF_NODISKIO
          SF_MNOWAIT
          SF_SYNC

   Parameters to the :func:`sendfile` function, if the implementation supports
   them.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1432)

Unknown directive type "data".

```
.. data:: SF_NOCACHE

   Parameter to the :func:`sendfile` function, if the implementation supports
   it. The data won't be cached in the virtual memory and will be freed afterwards.

   .. availability:: Unix.

   .. versionadded:: 3.11
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1442)

Unknown directive type "function".

```
.. function:: splice(src, dst, count, offset_src=None, offset_dst=None)

   Transfer *count* bytes from file descriptor *src*, starting from offset
   *offset_src*, to file descriptor *dst*, starting from offset *offset_dst*.
   At least one of the file descriptors must refer to a pipe. If *offset_src*
   is None, then *src* is read from the current position; respectively for
   *offset_dst*. The offset associated to the file descriptor that refers to a
   pipe must be ``None``. The files pointed by *src* and *dst* must reside in
   the same filesystem, otherwise an :exc:`OSError` is raised with
   :attr:`~OSError.errno` set to :data:`errno.EXDEV`.

   This copy is done without the additional cost of transferring data
   from the kernel to user space and then back into the kernel. Additionally,
   some filesystems could implement extra optimizations. The copy is done as if
```

```
      both files are opened as binary.

      Upon successful completion, returns the number of bytes spliced to or from
      the pipe. A return value of 0 means end of input. If *src* refers to a
      pipe, then this means that there was no data to transfer, and it would not
      make sense to block because there are no writers connected to the write end
      of the pipe.

      .. availability:: Linux kernel >= 2.6.17 and glibc >= 2.5

      .. versionadded:: 3.10
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1469)

Unknown directive type "data".

```
   .. data:: SPLICE_F_MOVE
             SPLICE_F_NONBLOCK
             SPLICE_F_MORE

      .. versionadded:: 3.10
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1475)

Unknown directive type "function".

```
   .. function:: readv(fd, buffers)

      Read from a file descriptor *fd* into a number of mutable :term:`bytes-like
      objects <bytes-like object>` *buffers*. Transfer data into each buffer until
      it is full and then move on to the next buffer in the sequence to hold the
      rest of the data.

      Return the total number of bytes actually read which can be less than the
      total capacity of all the objects.

      The operating system may set a limit (:func:`sysconf` value
      ``'SC_IOV_MAX'``) on the number of buffers that can be used.

      .. availability:: Unix.

      .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1493)

Unknown directive type "function".

```
   .. function:: tcgetpgrp(fd)

      Return the process group associated with the terminal given by *fd* (an open
      file descriptor as returned by :func:`os.open`).

      .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1501)

Unknown directive type "function".

```
   .. function:: tcsetpgrp(fd, pg)

      Set the process group associated with the terminal given by *fd* (an open file
      descriptor as returned by :func:`os.open`) to *pg*.

      .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1509)

Unknown directive type "function".

```
   .. function:: ttyname(fd)
   Return a string which specifies the terminal device associated with
   file descriptor *fd*.  If *fd* is not associated with a terminal device, an
   exception is raised.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1518)

Unknown directive type "function".

```
.. function:: write(fd, str)

   Write the bytestring in *str* to file descriptor *fd*.

   Return the number of bytes actually written.

   .. note::

      This function is intended for low-level I/O and must be applied to a file
      descriptor as returned by :func:`os.open` or :func:`pipe`.  To write a "file
      object" returned by the built-in function :func:`open` or by :func:`popen` or
      :func:`fdopen`, or :data:`sys.stdout` or :data:`sys.stderr`, use its
      :meth:`~file.write` method.

   .. versionchanged:: 3.5
      If the system call is interrupted and the signal handler does not raise an
      exception, the function now retries the system call instead of raising an
      :exc:`InterruptedError` exception (see :pep:`475` for the rationale).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1538)

Unknown directive type "function".

```
.. function:: writev(fd, buffers)

   Write the contents of *buffers* to file descriptor *fd*. *buffers* must be
   a sequence of :term:`bytes-like objects <bytes-like object>`. Buffers are
   processed in array order. Entire contents of the first buffer is written
   before proceeding to the second, and so on.

   Returns the total number of bytes actually written.

   The operating system may set a limit (:func:`sysconf` value
   ``'SC_IOV_MAX'``) on the number of buffers that can be used.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

Querying the size of a terminal

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1560)

Unknown directive type "versionadded".

```
.. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1562)

Unknown directive type "function".

```
.. function:: get_terminal_size(fd=STDOUT_FILENO)

   Return the size of the terminal window as ``(columns, lines)``,
   tuple of type :class:`terminal_size`.

   The optional argument ``fd`` (default ``STDOUT_FILENO``, or standard
   output) specifies which file descriptor should be queried.

   If the file descriptor is not connected to a terminal, an :exc:`OSError`
   is raised.
   :func:`shutil.get_terminal_size` is the high-level function which
   should normally be used, ``os.get_terminal_size`` is the low-level
   implementation.

   .. availability:: Unix, Windows.
```

A subclass of tuple, holding `(columns, lines)` of the terminal window size.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1583)

Unknown directive type "attribute".

```
.. attribute:: columns

   Width of the terminal window in characters.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1587)

Unknown directive type "attribute".

```
.. attribute:: lines

   Height of the terminal window in characters.
```

Inheritance of File Descriptors

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1597)

Unknown directive type "versionadded".

```
.. versionadded:: 3.4
```

A file descriptor has an "inheritable" flag which indicates if the file descriptor can be inherited by child processes. Since Python 3.4, file descriptors created by Python are non-inheritable by default.

On UNIX, non-inheritable file descriptors are closed in child processes at the execution of a new program, other file descriptors are inherited.

On Windows, non-inheritable handles and file descriptors are closed in child processes, except for standard streams (file descriptors 0, 1 and 2: stdin, stdout and stderr), which are always inherited. Using :func:`spawn\* <spawnl>` functions, all inheritable handles and all inheritable file descriptors are inherited. Using the :mod:`subprocess` module, all file descriptors except standard streams are closed, and inheritable handles are only inherited if the *close_fds* parameter is `False`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1606); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1606); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1614)

Unknown directive type "function".

```
.. function:: get_inheritable(fd)

   Get the "inheritable" flag of the specified file descriptor (a boolean).
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1618)

Unknown directive type "function".

```
.. function:: set_inheritable(fd, inheritable)

   Set the "inheritable" flag of the specified file descriptor.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1622)

Unknown directive type "function".

```
.. function:: get_handle_inheritable(handle)

   Get the "inheritable" flag of the specified handle (a boolean).

   .. availability:: Windows.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1628)

Unknown directive type "function".

```
.. function:: set_handle_inheritable(handle, inheritable)

   Set the "inheritable" flag of the specified handle.

   .. availability:: Windows.
```

Files and Directories

On some Unix platforms, many of these functions support one or more of these features:

- **specifying a file descriptor:** Normally the *path* argument provided to functions in the :mod:`os` module must be a string specifying a file path. However, some functions now alternatively accept an open file descriptor for their *path* argument. The function will then operate on the file referred to by the descriptor. (For POSIX systems, Python will call the variant of the function prefixed with `f` (e.g. call `fchdir` instead of `chdir`).)

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1645); *backlink*

  Unknown interpreted text role "mod".

  You can check whether or not *path* can be specified as a file descriptor for a particular function on your platform using :data:`os.supports_fd`. If this functionality is unavailable, using it will raise a :exc:`NotImplementedError`.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1653); *backlink*

  Unknown interpreted text role "data".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1653); *backlink*

  Unknown interpreted text role "exc".

  If the function also supports *dir_fd* or *follow_symlinks* arguments, it's an error to specify one of those when supplying *path* as a file descriptor.

- **paths relative to directory descriptors:** If *dir_fd* is not `None`, it should be a file descriptor referring to a directory, and the path to operate on should be relative; path will then be relative to that directory. If the path is absolute, *dir_fd* is ignored. (For POSIX systems, Python will call the variant of the function with an `at` suffix and possibly prefixed with `f` (e.g. call `faccessat` instead of `access`).

  You can check whether or not *dir_fd* is supported for a particular function on your platform using :data:`os.supports_dir_fd`. If it's unavailable, using it will raise a :exc:`NotImplementedError`.

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1670); *backlink*

  Unknown interpreted text role "data".

  System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1670); *backlink*

Unknown interpreted text role "exc".

- **not following symlinks:** If *follow_symlinks* is `False`, and the last element of the path to operate on is a symbolic link, the function will operate on the symbolic link itself rather than the file pointed to by the link. (For POSIX systems, Python will call the `l...` variant of the function.)

  You can check whether or not *follow_symlinks* is supported for a particular function on your platform using :data:`os.supports_follow_symlinks`. If it's unavailable, using it will raise a :exc:`NotImplementedError`.

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1682); *backlink*

  Unknown interpreted text role "data".

  System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1682); *backlink*

  Unknown interpreted text role "exc".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1688)

Unknown directive type "function".

```
.. function:: access(path, mode, *, dir_fd=None, effective_ids=False, follow_symlinks=True)

   Use the real uid/gid to test for access to *path*.  Note that most operations
   will use the effective uid/gid, therefore this routine can be used in a
   suid/sgid environment to test if the invoking user has the specified access to
   *path*.  *mode* should be :const:`F_OK` to test the existence of *path*, or it
   can be the inclusive OR of one or more of :const:`R_OK`, :const:`W_OK`, and
   :const:`X_OK` to test permissions.  Return :const:`True` if access is allowed,
   :const:`False` if not. See the Unix man page :manpage:`access(2)` for more
   information.

   This function can support specifying :ref:`paths relative to directory
   descriptors <dir_fd>` and :ref:`not following symlinks <follow_symlinks>`.

   If *effective_ids* is ``True``, :func:`access` will perform its access
   checks using the effective uid/gid instead of the real uid/gid.
   *effective_ids* may not be supported on your platform; you can check whether
   or not it is available using :data:`os.supports_effective_ids`.  If it is
   unavailable, using it will raise a :exc:`NotImplementedError`.

   .. note::

      Using :func:`access` to check if a user is authorized to e.g. open a file
      before actually doing so using :func:`open` creates a security hole,
      because the user might exploit the short time interval between checking
      and opening the file to manipulate it. It's preferable to use :term:`EAFP`
      techniques. For example::

         if os.access("myfile", os.R_OK):
             with open("myfile") as fp:
                 return fp.read()
         return "some default data"

      is better written as::

         try:
             fp = open("myfile")
         except PermissionError:
             return "some default data"
         else:
             with fp:
                 return fp.read()

   .. note::

      I/O operations may fail even when :func:`access` indicates that they would
      succeed, particularly for operations on network filesystems which may have
      permissions semantics beyond the usual POSIX permission-bit model.

   .. versionchanged:: 3.3
      Added the *dir_fd*, *effective_ids*, and *follow_symlinks* parameters.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1744)

Unknown directive type "data".

```
.. data:: F_OK
          R_OK
          W_OK
          X_OK

   Values to pass as the *mode* parameter of :func:`access` to test the
   existence, readability, writability and executability of *path*,
   respectively.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1754)

Unknown directive type "function".

```
.. function:: chdir(path)

   .. index:: single: directory; changing

   Change the current working directory to *path*.

   This function can support :ref:`specifying a file descriptor <path_fd>`.  The
   descriptor must refer to an opened directory, not an open file.

   This function can raise :exc:`OSError` and subclasses such as
   :exc:`FileNotFoundError`, :exc:`PermissionError`, and :exc:`NotADirectoryError`.

   .. audit-event:: os.chdir path os.chdir

   .. versionadded:: 3.3
      Added support for specifying *path* as a file descriptor
      on some platforms.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1776)

Unknown directive type "function".

```
.. function:: chflags(path, flags, *, follow_symlinks=True)

   Set the flags of *path* to the numeric *flags*. *flags* may take a combination
   (bitwise OR) of the following values (as defined in the :mod:`stat` module):

   * :data:`stat.UF_NODUMP`
   * :data:`stat.UF_IMMUTABLE`
   * :data:`stat.UF_APPEND`
   * :data:`stat.UF_OPAQUE`
   * :data:`stat.UF_NOUNLINK`
   * :data:`stat.UF_COMPRESSED`
   * :data:`stat.UF_HIDDEN`
   * :data:`stat.SF_ARCHIVED`
   * :data:`stat.SF_IMMUTABLE`
   * :data:`stat.SF_APPEND`
   * :data:`stat.SF_NOUNLINK`
   * :data:`stat.SF_SNAPSHOT`

   This function can support :ref:`not following symlinks <follow_symlinks>`.

   .. audit-event:: os.chflags path,flags os.chflags

   .. availability:: Unix.

   .. versionadded:: 3.3
      The *follow_symlinks* argument.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1807)

Unknown directive type "function".

```
.. function:: chmod(path, mode, *, dir_fd=None, follow_symlinks=True)

   Change the mode of *path* to the numeric *mode*. *mode* may take one of the
   following values (as defined in the :mod:`stat` module) or bitwise ORed
   combinations of them:

   * :data:`stat.S_ISUID`
   * :data:`stat.S_ISGID`
   * :data:`stat.S_ENFMT`
   * :data:`stat.S_ISVTX`
   * :data:`stat.S_IREAD`
   * :data:`stat.S_IWRITE`
   * :data:`stat.S_IEXEC`
   * :data:`stat.S_IRWXU`
   * :data:`stat.S_IRUSR`
   * :data:`stat.S_IWUSR`
   * :data:`stat.S_IXUSR`
   * :data:`stat.S_IRWXG`
   * :data:`stat.S_IRGRP`
   * :data:`stat.S_IWGRP`
   * :data:`stat.S_IXGRP`
   * :data:`stat.S_IRWXO`
   * :data:`stat.S_IROTH`
   * :data:`stat.S_IWOTH`
   * :data:`stat.S_IXOTH`

   This function can support :ref:`specifying a file descriptor <path_fd>`,
   :ref:`paths relative to directory descriptors <dir_fd>` and :ref:`not
   following symlinks <follow_symlinks>`.

   .. note::

      Although Windows supports :func:`chmod`, you can only set the file's
      read-only flag with it (via the ``stat.S_IWRITE`` and ``stat.S_IREAD``
      constants or a corresponding integer value).  All other bits are ignored.

   .. audit-event:: os.chmod path,mode,dir_fd os.chmod

   .. versionadded:: 3.3
      Added support for specifying *path* as an open file descriptor,
      and the *dir_fd* and *follow_symlinks* arguments.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1853)

Unknown directive type "function".

```
.. function:: chown(path, uid, gid, *, dir_fd=None, follow_symlinks=True)

   Change the owner and group id of *path* to the numeric *uid* and *gid*.  To
   leave one of the ids unchanged, set it to -1.

   This function can support :ref:`specifying a file descriptor <path_fd>`,
   :ref:`paths relative to directory descriptors <dir_fd>` and :ref:`not
   following symlinks <follow_symlinks>`.

   See :func:`shutil.chown` for a higher-level function that accepts names in
   addition to numeric ids.

   .. audit-event:: os.chown path,uid,gid,dir_fd os.chown

   .. availability:: Unix.

   .. versionadded:: 3.3
      Added support for specifying *path* as an open file descriptor,
      and the *dir_fd* and *follow_symlinks* arguments.

   .. versionchanged:: 3.6
      Supports a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1877)

Unknown directive type "function".

```
.. function:: chroot(path)

   Change the root directory of the current process to *path*.

   .. availability:: Unix.
   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1887)

Unknown directive type "function".

```
.. function:: fchdir(fd)

   Change the current working directory to the directory represented by the file
   descriptor *fd*.  The descriptor must refer to an opened directory, not an
   open file.  As of Python 3.3, this is equivalent to ``os.chdir(fd)``.

   .. audit-event:: os.chdir path os.fchdir

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1898)

Unknown directive type "function".

```
.. function:: getcwd()

   Return a string representing the current working directory.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1903)

Unknown directive type "function".

```
.. function:: getcwdb()

   Return a bytestring representing the current working directory.

   .. versionchanged:: 3.8
      The function now uses the UTF-8 encoding on Windows, rather than the ANSI
      code page: see :pep:`529` for the rationale. The function is no longer
      deprecated on Windows.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1913)

Unknown directive type "function".

```
.. function:: lchflags(path, flags)

   Set the flags of *path* to the numeric *flags*, like :func:`chflags`, but do
   not follow symbolic links.  As of Python 3.3, this is equivalent to
   ``os.chflags(path, flags, follow_symlinks=False)``.

   .. audit-event:: os.chflags path,flags os.lchflags

   .. availability:: Unix.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 1927)

Unknown directive type "function".

```
.. function:: lchmod(path, mode)

   Change the mode of *path* to the numeric *mode*. If path is a symlink, this
   affects the symlink rather than the target.  See the docs for :func:`chmod`
   for possible values of *mode*.  As of Python 3.3, this is equivalent to
   ``os.chmod(path, mode, follow_symlinks=False)``.

   .. audit-event:: os.chmod path,mode,dir_fd os.lchmod
   .. availability:: Unix.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1941)

Unknown directive type "function".

```
.. function:: lchown(path, uid, gid)

   Change the owner and group id of *path* to the numeric *uid* and *gid*.  This
   function will not follow symbolic links.  As of Python 3.3, this is equivalent
   to ``os.chown(path, uid, gid, follow_symlinks=False)``.

   .. audit-event:: os.chown path,uid,gid,dir_fd os.lchown

   .. availability:: Unix.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1955)

Unknown directive type "function".

```
.. function:: link(src, dst, *, src_dir_fd=None, dst_dir_fd=None, follow_symlinks=True)

   Create a hard link pointing to *src* named *dst*.

   This function can support specifying *src_dir_fd* and/or *dst_dir_fd* to
   supply :ref:`paths relative to directory descriptors <dir_fd>`, and :ref:`not
   following symlinks <follow_symlinks>`.

   .. audit-event:: os.link src,dst,src_dir_fd,dst_dir_fd os.link

   .. availability:: Unix, Windows.

   .. versionchanged:: 3.2
      Added Windows support.

   .. versionadded:: 3.3
      Added the *src_dir_fd*, *dst_dir_fd*, and *follow_symlinks* arguments.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object` for *src* and *dst*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 1977)

Unknown directive type "function".

```
.. function:: listdir(path='.')

   Return a list containing the names of the entries in the directory given by
   *path*.  The list is in arbitrary order, and does not include the special
   entries ``'.'`` and ``'..'`` even if they are present in the directory.
   If a file is removed from or added to the directory during the call of
   this function, whether a name for that file be included is unspecified.

   *path* may be a :term:`path-like object`.  If *path* is of type ``bytes``
   (directly or indirectly through the :class:`PathLike` interface),
   the filenames returned will also be of type ``bytes``;
   in all other circumstances, they will be of type ``str``.

   This function can also support :ref:`specifying a file descriptor
   <path_fd>`; the file descriptor must refer to a directory.

   .. audit-event:: os.listdir path os.listdir

   .. note::
      To encode ``str`` filenames to ``bytes``, use :func:`~os.fsencode`.

   .. seealso::

      The :func:`scandir` function returns directory entries along with
      file attribute information, giving better performance for many
      common use cases.
   .. versionchanged:: 3.2
      The *path* parameter became optional.

   .. versionadded:: 3.3
      Added support for specifying *path* as an open file descriptor.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2014)

Unknown directive type "function".

```
.. function:: lstat(path, *, dir_fd=None)

   Perform the equivalent of an :c:func:`lstat` system call on the given path.
   Similar to :func:`~os.stat`, but does not follow symbolic links. Return a
   :class:`stat_result` object.

   On platforms that do not support symbolic links, this is an alias for
   :func:`~os.stat`.

   As of Python 3.3, this is equivalent to ``os.stat(path, dir_fd=dir_fd,
   follow_symlinks=False)``.

   This function can also support :ref:`paths relative to directory descriptors
   <dir_fd>`.

   .. seealso::

      The :func:`.stat` function.

   .. versionchanged:: 3.2
      Added support for Windows 6.0 (Vista) symbolic links.

   .. versionchanged:: 3.3
      Added the *dir_fd* parameter.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.

   .. versionchanged:: 3.8
      On Windows, now opens reparse points that represent another path
      (name surrogates), including symbolic links and directory junctions.
      Other kinds of reparse points are resolved by the operating system as
      for :func:`~os.stat`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2049)

Unknown directive type "function".

```
.. function:: mkdir(path, mode=0o777, *, dir_fd=None)

   Create a directory named *path* with numeric mode *mode*.

   If the directory already exists, :exc:`FileExistsError` is raised. If a parent
   directory in the path does not exist, :exc:`FileNotFoundError` is raised.

   .. _mkdir_modebits:

   On some systems, *mode* is ignored.  Where it is used, the current umask
   value is first masked out.  If bits other than the last 9 (i.e. the last 3
   digits of the octal representation of the *mode*) are set, their meaning is
   platform-dependent.  On some platforms, they are ignored and you should call
   :func:`chmod` explicitly to set them.

   This function can also support :ref:`paths relative to directory descriptors
   <dir_fd>`.

   It is also possible to create temporary directories; see the
   :mod:`tempfile` module's :func:`tempfile.mkdtemp` function.

   .. audit-event:: os.mkdir path,mode,dir_fd os.mkdir

   .. versionadded:: 3.3
      The *dir_fd* argument.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2079)

Unknown directive type "function".

```
.. function:: makedirs(name, mode=0o777, exist_ok=False)

   .. index::
      single: directory; creating
      single: UNC paths; and os.makedirs()

   Recursive directory creation function.  Like :func:`mkdir`, but makes all
   intermediate-level directories needed to contain the leaf directory.

   The *mode* parameter is passed to :func:`mkdir` for creating the leaf
   directory; see :ref:`the mkdir() description <mkdir_modebits>` for how it
   is interpreted.  To set the file permission bits of any newly-created parent
   directories you can set the umask before invoking :func:`makedirs`.  The
   file permission bits of existing parent directories are not changed.

   If *exist_ok* is ``False`` (the default), a :exc:`FileExistsError` is
   raised if the target directory already exists.

   .. note::

      :func:`makedirs` will become confused if the path elements to create
      include :data:`pardir` (eg. ".." on UNIX systems).

   This function handles UNC paths correctly.

   .. audit-event:: os.mkdir path,mode,dir_fd os.makedirs

   .. versionadded:: 3.2
      The *exist_ok* parameter.

   .. versionchanged:: 3.4.1

      Before Python 3.4.1, if *exist_ok* was ``True`` and the directory existed,
      :func:`makedirs` would still raise an error if *mode* did not match the
      mode of the existing directory. Since this behavior was impossible to
      implement safely, it was removed in Python 3.4.1. See :issue:`21082`.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.

   .. versionchanged:: 3.7
      The *mode* argument no longer affects the file permission bits of
      newly-created intermediate-level directories.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2124)

Unknown directive type "function".

```
.. function:: mkfifo(path, mode=0o666, *, dir_fd=None)

   Create a FIFO (a named pipe) named *path* with numeric mode *mode*.
   The current umask value is first masked out from the mode.

   This function can also support :ref:`paths relative to directory descriptors
   <dir_fd>`.

   FIFOs are pipes that can be accessed like regular files.  FIFOs exist until they
   are deleted (for example with :func:`os.unlink`). Generally, FIFOs are used as
   rendezvous between "client" and "server" type processes: the server opens the
   FIFO for reading, and the client opens it for writing.  Note that :func:`mkfifo`
   doesn't open the FIFO --- it just creates the rendezvous point.

   .. availability:: Unix.

   .. versionadded:: 3.3
      The *dir_fd* argument.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2147)

Unknown directive type "function".

```
.. function:: mknod(path, mode=0o600, device=0, *, dir_fd=None)

   Create a filesystem node (file, device special file or named pipe) named
   *path*. *mode* specifies both the permissions to use and the type of node
   to be created, being combined (bitwise OR) with one of ``stat.S_IFREG``,
   ``stat.S_IFCHR``, ``stat.S_IFBLK``, and ``stat.S_IFIFO`` (those constants are
   available in :mod:`stat`).  For ``stat.S_IFCHR`` and ``stat.S_IFBLK``,
   *device* defines the newly created device special file (probably using
   :func:`os.makedev`), otherwise it is ignored.

   This function can also support :ref:`paths relative to directory descriptors
   <dir_fd>`.

   .. availability:: Unix.

   .. versionadded:: 3.3
      The *dir_fd* argument.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2169)

Unknown directive type "function".

```
.. function:: major(device)

   Extract the device major number from a raw device number (usually the
   :attr:`st_dev` or :attr:`st_rdev` field from :c:type:`stat`).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2175)

Unknown directive type "function".

```
.. function:: minor(device)

   Extract the device minor number from a raw device number (usually the
   :attr:`st_dev` or :attr:`st_rdev` field from :c:type:`stat`).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2181)

Unknown directive type "function".

```
.. function:: makedev(major, minor)

   Compose a raw device number from the major and minor device numbers.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2186)

Unknown directive type "function".

```
.. function:: pathconf(path, name)

   Return system configuration information relevant to a named file. *name*
   specifies the configuration value to retrieve; it may be a string which is the
   name of a defined system value; these names are specified in a number of
   standards (POSIX.1, Unix 95, Unix 98, and others).  Some platforms define
   additional names as well.  The names known to the host operating system are
   given in the ``pathconf_names`` dictionary.  For configuration variables not
   included in that mapping, passing an integer for *name* is also accepted.

   If *name* is a string and is not known, :exc:`ValueError` is raised.  If a
   specific value for *name* is not supported by the host system, even if it is
   included in ``pathconf_names``, an :exc:`OSError` is raised with
   :const:`errno.EINVAL` for the error number.

   This function can support :ref:`specifying a file descriptor
   <path_fd>`.

   .. availability:: Unix.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2210)

Unknown directive type "data".

```
.. data:: pathconf_names

   Dictionary mapping names accepted by :func:`pathconf` and :func:`fpathconf` to
   the integer values defined for those names by the host operating system.  This
   can be used to determine the set of names known to the system.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2219)

Unknown directive type "function".

```
.. function:: readlink(path, *, dir_fd=None)

   Return a string representing the path to which the symbolic link points.  The
   result may be either an absolute or relative pathname; if it is relative, it
   may be converted to an absolute pathname using
   ``os.path.join(os.path.dirname(path), result)``.

   If the *path* is a string object (directly or indirectly through a
   :class:`PathLike` interface), the result will also be a string object,
   and the call may raise a UnicodeDecodeError. If the *path* is a bytes
   object (direct or indirectly), the result will be a bytes object.

   This function can also support :ref:`paths relative to directory descriptors
   <dir_fd>`.

   When trying to resolve a path that may contain links, use
   :func:`~os.path.realpath` to properly handle recursion and platform
   differences.

   .. availability:: Unix, Windows.

   .. versionchanged:: 3.2
      Added support for Windows 6.0 (Vista) symbolic links.

   .. versionadded:: 3.3
      The *dir_fd* argument.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object` on Unix.

   .. versionchanged:: 3.8
      Accepts a :term:`path-like object` and a bytes object on Windows.

   .. versionchanged:: 3.8
      Added support for directory junctions, and changed to return the
      substitution path (which typically includes ``\\?\`` prefix) rather
      than the optional "print name" field that was previously returned.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2257)

Unknown directive type "function".

```
.. function:: remove(path, *, dir_fd=None)

   Remove (delete) the file *path*.  If *path* is a directory, an
   :exc:`IsADirectoryError` is raised.  Use :func:`rmdir` to remove directories.
   If the file does not exist, a :exc:`FileNotFoundError` is raised.

   This function can support :ref:`paths relative to directory descriptors
   <dir_fd>`.

   On Windows, attempting to remove a file that is in use causes an exception to
   be raised; on Unix, the directory entry is removed but the storage allocated
   to the file is not made available until the original file is no longer in use.

   This function is semantically identical to :func:`unlink`.

   .. audit-event:: os.remove path,dir_fd os.remove

   .. versionadded:: 3.3
      The *dir_fd* argument.
   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2281)

Unknown directive type "function".

```
.. function:: removedirs(name)

   .. index:: single: directory; deleting

   Remove directories recursively.  Works like :func:`rmdir` except that, if the
   leaf directory is successfully removed, :func:`removedirs`  tries to
   successively remove every parent directory mentioned in  *path* until an error
   is raised (which is ignored, because it generally means that a parent directory
   is not empty). For example, ``os.removedirs('foo/bar/baz')`` will first remove
   the directory ``'foo/bar/baz'``, and then remove ``'foo/bar'`` and ``'foo'`` if
   they are empty. Raises :exc:`OSError` if the leaf directory could not be
   successfully removed.

   .. audit-event:: os.remove path,dir_fd os.removedirs

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2300)

Unknown directive type "function".

```
.. function:: rename(src, dst, *, src_dir_fd=None, dst_dir_fd=None)

   Rename the file or directory *src* to *dst*. If *dst* exists, the operation
   will fail with an :exc:`OSError` subclass in a number of cases:

   On Windows, if *dst* exists a :exc:`FileExistsError` is always raised.

   On Unix, if *src* is a file and *dst* is a directory or vice-versa, an
   :exc:`IsADirectoryError` or a :exc:`NotADirectoryError` will be raised
   respectively.  If both are directories and *dst* is empty, *dst* will be
   silently replaced.  If *dst* is a non-empty directory, an :exc:`OSError`
   is raised. If both are files, *dst* it will be replaced silently if the user
   has permission.  The operation may fail on some Unix flavors if *src* and
   *dst* are on different filesystems.  If successful, the renaming will be an
   atomic operation (this is a POSIX requirement).

   This function can support specifying *src_dir_fd* and/or *dst_dir_fd* to
   supply :ref:`paths relative to directory descriptors <dir_fd>`.

   If you want cross-platform overwriting of the destination, use :func:`replace`.

   .. audit-event:: os.rename src,dst,src_dir_fd,dst_dir_fd os.rename

   .. versionadded:: 3.3
      The *src_dir_fd* and *dst_dir_fd* arguments.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object` for *src* and *dst*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2330)

Unknown directive type "function".

```
.. function:: renames(old, new)

   Recursive directory or file renaming function. Works like :func:`rename`, except
   creation of any intermediate directories needed to make the new pathname good is
   attempted first. After the rename, directories corresponding to rightmost path
   segments of the old name will be pruned away using :func:`removedirs`.

   .. note::

      This function can fail with the new directory structure made if you lack
      permissions needed to remove the leaf directory or file.

   .. audit-event:: os.rename src,dst,src_dir_fd,dst_dir_fd os.renames
   .. versionchanged:: 3.6
      Accepts a :term:`path-like object` for *old* and *new*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2348)

Unknown directive type "function".

```
.. function:: replace(src, dst, *, src_dir_fd=None, dst_dir_fd=None)

   Rename the file or directory *src* to *dst*.  If *dst* is a directory,
   :exc:`OSError` will be raised.  If *dst* exists and is a file, it will
   be replaced silently if the user has permission.  The operation may fail
   if *src* and *dst* are on different filesystems.  If successful,
   the renaming will be an atomic operation (this is a POSIX requirement).

   This function can support specifying *src_dir_fd* and/or *dst_dir_fd* to
   supply :ref:`paths relative to directory descriptors <dir_fd>`.

   .. audit-event:: os.rename src,dst,src_dir_fd,dst_dir_fd os.replace

   .. versionadded:: 3.3

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object` for *src* and *dst*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2367)

Unknown directive type "function".

```
.. function:: rmdir(path, *, dir_fd=None)

   Remove (delete) the directory *path*.  If the directory does not exist or is
   not empty, a :exc:`FileNotFoundError` or an :exc:`OSError` is raised
   respectively.  In order to remove whole directory trees,
   :func:`shutil.rmtree` can be used.

   This function can support :ref:`paths relative to directory descriptors
   <dir_fd>`.

   .. audit-event:: os.rmdir path,dir_fd os.rmdir

   .. versionadded:: 3.3
      The *dir_fd* parameter.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2386)

Unknown directive type "function".

```
.. function:: scandir(path='.')

   Return an iterator of :class:`os.DirEntry` objects corresponding to the
   entries in the directory given by *path*. The entries are yielded in
   arbitrary order, and the special entries ``'.'`` and ``'..'`` are not
   included.  If a file is removed from or added to the directory after
   creating the iterator, whether an entry for that file be included is
   unspecified.

   Using :func:`scandir` instead of :func:`listdir` can significantly
   increase the performance of code that also needs file type or file
   attribute information, because :class:`os.DirEntry` objects expose this
   information if the operating system provides it when scanning a directory.
   All :class:`os.DirEntry` methods may perform a system call, but
   :func:`~os.DirEntry.is_dir` and :func:`~os.DirEntry.is_file` usually only
   require a system call for symbolic links; :func:`os.DirEntry.stat`
   always requires a system call on Unix but only requires one for
   symbolic links on Windows.

   *path* may be a :term:`path-like object`.  If *path* is of type ``bytes``
   (directly or indirectly through the :class:`PathLike` interface),
   the type of the :attr:`~os.DirEntry.name` and :attr:`~os.DirEntry.path`
   attributes of each :class:`os.DirEntry` will be ``bytes``; in all other
   circumstances, they will be of type ``str``.

   This function can also support :ref:`specifying a file descriptor
```

```
<path_fd>`; the file descriptor must refer to a directory.

.. audit-event:: os.scandir path os.scandir

The :func:`scandir` iterator supports the :term:`context manager` protocol
and has the following method:

.. method:: scandir.close()

   Close the iterator and free acquired resources.

   This is called automatically when the iterator is exhausted or garbage
   collected, or when an error happens during iterating.  However it
   is advisable to call it explicitly or use the :keyword:`with`
   statement.

   .. versionadded:: 3.6

The following example shows a simple use of :func:`scandir` to display all
the files (excluding directories) in the given *path* that don't start with
``'.'``. The ``entry.is_file()`` call will generally not make an additional
system call::

   with os.scandir(path) as it:
       for entry in it:
           if not entry.name.startswith('.') and entry.is_file():
               print(entry.name)

.. note::

   On Unix-based systems, :func:`scandir` uses the system's
   `opendir() <http://pubs.opengroup.org/onlinepubs/009695399/functions/opendir.html>`_
   and
   `readdir() <http://pubs.opengroup.org/onlinepubs/009695399/functions/readdir_r.html>`_
   functions. On Windows, it uses the Win32
   `FindFirstFileW <https://msdn.microsoft.com/en-us/library/windows/desktop/aa364418(v=vs.85).aspx
   and
   `FindNextFileW <https://msdn.microsoft.com/en-us/library/windows/desktop/aa364428(v=vs.85).aspx>
   functions.

.. versionadded:: 3.5

.. versionadded:: 3.6
   Added support for the :term:`context manager` protocol and the
   :func:`~scandir.close()` method.  If a :func:`scandir` iterator is neither
   exhausted nor explicitly closed a :exc:`ResourceWarning` will be emitted
   in its destructor.

   The function accepts a :term:`path-like object`.

.. versionchanged:: 3.7
   Added support for :ref:`file descriptors <path_fd>` on Unix.
```

Object yielded by :func:`scandir` to expose the file path and other file attributes of a directory entry.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2468); *backlink*

Unknown interpreted text role "func".

:func:`scandir` will provide as much of this information as possible without making additional system calls. When a `stat()` or `lstat()` system call is made, the `os.DirEntry` object will cache the result.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2471); *backlink*

Unknown interpreted text role "func".

`os.DirEntry` instances are not intended to be stored in long-lived data structures; if you know the file metadata has changed or if a long time has elapsed since calling :func:`scandir`, call `os.stat(entry.path)` to fetch up-to-date information.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2475); *backlink*

Unknown interpreted text role "func".

Because the `os.DirEntry` methods can make operating system calls, they may also raise :exc:`OSError`. If you need very fine-grained control over errors, you can catch :exc:`OSError` when calling one of the `os.DirEntry` methods and handle as appropriate.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2480); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2480); *backlink*

Unknown interpreted text role "exc".

To be directly usable as a :term:`path-like object`, `os.DirEntry` implements the :class:`PathLike` interface.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2485); *backlink*

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2485); *backlink*

Unknown interpreted text role "class".

Attributes and methods on a `os.DirEntry` instance are as follows:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2490)

Unknown directive type "attribute".

```
.. attribute:: name

   The entry's base filename, relative to the :func:`scandir` *path*
   argument.

   The :attr:`name` attribute will be ``bytes`` if the :func:`scandir`
   *path* argument is of type ``bytes`` and ``str`` otherwise.  Use
   :func:`~os.fsdecode` to decode byte filenames.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2499)

Unknown directive type "attribute".

```
.. attribute:: path

   The entry's full path name: equivalent to ``os.path.join(scandir_path,
   entry.name)`` where *scandir_path* is the :func:`scandir` *path*
   argument.  The path is only absolute if the :func:`scandir` *path*
   argument was absolute.  If the :func:`scandir` *path*
   argument was a :ref:`file descriptor <path_fd>`, the :attr:`path`
   attribute is the same as the :attr:`name` attribute.

   The :attr:`path` attribute will be ``bytes`` if the :func:`scandir`
   *path* argument is of type ``bytes`` and ``str`` otherwise.  Use
   :func:`~os.fsdecode` to decode byte filenames.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2512)

Unknown directive type "method".

```
.. method:: inode()

   Return the inode number of the entry.

   The result is cached on the ``os.DirEntry`` object. Use
   ``os.stat(entry.path, follow_symlinks=False).st_ino`` to fetch up-to-date
   information.

   On the first, uncached call, a system call is required on Windows but
   not on Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2523)

Unknown directive type "method".

```
.. method:: is_dir(*, follow_symlinks=True)
```

```
      Return ``True`` if this entry is a directory or a symbolic link pointing
      to a directory; return ``False`` if the entry is or points to any other
      kind of file, or if it doesn't exist anymore.

      If *follow_symlinks* is ``False``, return ``True`` only if this entry
      is a directory (without following symlinks); return ``False`` if the
      entry is any other kind of file or if it doesn't exist anymore.

      The result is cached on the ``os.DirEntry`` object, with a separate cache
      for *follow_symlinks* ``True`` and ``False``. Call :func:`os.stat` along
      with :func:`stat.S_ISDIR` to fetch up-to-date information.

      On the first, uncached call, no system call is required in most cases.
      Specifically, for non-symlinks, neither Windows or Unix require a system
      call, except on certain Unix file systems, such as network file systems,
      that return ``dirent.d_type == DT_UNKNOWN``. If the entry is a symlink,
      a system call will be required to follow the symlink unless
      *follow_symlinks* is ``False``.

      This method can raise :exc:`OSError`, such as :exc:`PermissionError`,
      but :exc:`FileNotFoundError` is caught and not raised.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2547)

Unknown directive type "method".

```
   .. method:: is_file(*, follow_symlinks=True)

      Return ``True`` if this entry is a file or a symbolic link pointing to a
      file; return ``False`` if the entry is or points to a directory or other
      non-file entry, or if it doesn't exist anymore.

      If *follow_symlinks* is ``False``, return ``True`` only if this entry
      is a file (without following symlinks); return ``False`` if the entry is
      a directory or other non-file entry, or if it doesn't exist anymore.

      The result is cached on the ``os.DirEntry`` object. Caching, system calls
      made, and exceptions raised are as per :func:`~os.DirEntry.is_dir`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2560)

Unknown directive type "method".

```
   .. method:: is_symlink()

      Return ``True`` if this entry is a symbolic link (even if broken);
      return ``False`` if the entry points to a directory or any kind of file,
      or if it doesn't exist anymore.

      The result is cached on the ``os.DirEntry`` object. Call
      :func:`os.path.islink` to fetch up-to-date information.

      On the first, uncached call, no system call is required in most cases.
      Specifically, neither Windows or Unix require a system call, except on
      certain Unix file systems, such as network file systems, that return
      ``dirent.d_type == DT_UNKNOWN``.

      This method can raise :exc:`OSError`, such as :exc:`PermissionError`,
      but :exc:`FileNotFoundError` is caught and not raised.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2577)

Unknown directive type "method".

```
   .. method:: stat(*, follow_symlinks=True)

      Return a :class:`stat_result` object for this entry. This method
      follows symbolic links by default; to stat a symbolic link add the
      ``follow_symlinks=False`` argument.

      On Unix, this method always requires a system call. On Windows, it
      only requires a system call if *follow_symlinks* is ``True`` and the
      entry is a reparse point (for example, a symbolic link or directory
      junction).

      On Windows, the ``st_ino``, ``st_dev`` and ``st_nlink`` attributes of the
      :class:`stat_result` are always set to zero. Call :func:`os.stat` to
      get these attributes.
      The result is cached on the ``os.DirEntry`` object, with a separate cache
      for *follow_symlinks* ``True`` and ``False``. Call :func:`os.stat` to
      fetch up-to-date information.
```

Note that there is a nice correspondence between several attributes and methods of `os.DirEntry` and of :class:`pathlib.Path`. In particular, the `name` attribute has the same meaning, as do the `is_dir()`, `is_file()`, `is_symlink()` and `stat()` methods.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2596); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2602)

Unknown directive type "versionadded".

```
.. versionadded:: 3.5
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2604)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.6
   Added support for the :class:`~os.PathLike` interface.  Added support
   for :class:`bytes` paths on Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2609)

Unknown directive type "function".

```
.. function:: stat(path, *, dir_fd=None, follow_symlinks=True)

   Get the status of a file or a file descriptor. Perform the equivalent of a
   :c:func:`stat` system call on the given path. *path* may be specified as
   either a string or bytes -- directly or indirectly through the :class:`PathLike`
   interface -- or as an open file descriptor. Return a :class:`stat_result`
   object.

   This function normally follows symlinks; to stat a symlink add the argument
   ``follow_symlinks=False``, or use :func:`lstat`.

   This function can support :ref:`specifying a file descriptor <path_fd>` and
   :ref:`not following symlinks <follow_symlinks>`.

   On Windows, passing ``follow_symlinks=False`` will disable following all
   name-surrogate reparse points, which includes symlinks and directory
   junctions. Other types of reparse points that do not resemble links or that
   the operating system is unable to follow will be opened directly. When
   following a chain of multiple links, this may result in the original link
   being returned instead of the non-link that prevented full traversal. To
   obtain stat results for the final path in this case, use the
   :func:`os.path.realpath` function to resolve the path name as far as
   possible and call :func:`lstat` on the result. This does not apply to
   dangling symlinks or junction points, which will raise the usual exceptions.

   .. index:: module: stat

   Example::

      >>> import os
      >>> statinfo = os.stat('somefile.txt')
      >>> statinfo
      os.stat_result(st_mode=33188, st_ino=7876932, st_dev=234881026,
      st_nlink=1, st_uid=501, st_gid=501, st_size=264, st_atime=1297230295,
      st_mtime=1297230027, st_ctime=1297230027)
      >>> statinfo.st_size
      264

   .. seealso::

      :func:`fstat` and :func:`lstat` functions.

   .. versionadded:: 3.3
      Added the *dir_fd* and *follow_symlinks* arguments, specifying a file
      descriptor instead of a path.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
   .. versionchanged:: 3.8
      On Windows, all reparse points that can be resolved by the operating
      system are now followed, and passing ``follow_symlinks=False``
      disables following all name surrogate reparse points. If the operating
      system reaches a reparse point that it is not able to follow, *stat* now
      returns the information for the original path as if
      ``follow_symlinks=False`` had been specified instead of raising an error.
```

Object whose attributes correspond roughly to the members of the :c:type:`stat` structure. It is used for the result of :func:`os.stat`, :func:`os.fstat` and :func:`os.lstat`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2669); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2669); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2669); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2669); *backlink*

Unknown interpreted text role "func".

Attributes:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2675)

Unknown directive type "attribute".

```
.. attribute:: st_mode

   File mode: file type and file mode bits (permissions).
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2679)

Unknown directive type "attribute".

```
.. attribute:: st_ino

   Platform dependent, but if non-zero, uniquely identifies the
   file for a given value of ``st_dev``. Typically:

   * the inode number on Unix,
   * the `file index
     <https://msdn.microsoft.com/en-us/library/aa363788>`_ on
     Windows
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2689)

Unknown directive type "attribute".

```
.. attribute:: st_dev

   Identifier of the device on which this file resides.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2693)

Unknown directive type "attribute".

```
.. attribute:: st_nlink
```

```
   Number of hard links.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2697)

Unknown directive type "attribute".

```
.. attribute:: st_uid

   User identifier of the file owner.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2701)

Unknown directive type "attribute".

```
.. attribute:: st_gid

   Group identifier of the file owner.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2705)

Unknown directive type "attribute".

```
.. attribute:: st_size

   Size of the file in bytes, if it is a regular file or a symbolic link.
   The size of a symbolic link is the length of the pathname it contains,
   without a terminating null byte.
```

Timestamps:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2713)

Unknown directive type "attribute".

```
.. attribute:: st_atime

   Time of most recent access expressed in seconds.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2717)

Unknown directive type "attribute".

```
.. attribute:: st_mtime

   Time of most recent content modification expressed in seconds.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2721)

Unknown directive type "attribute".

```
.. attribute:: st_ctime

   Platform dependent:

   * the time of most recent metadata change on Unix,
   * the time of creation on Windows, expressed in seconds.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2728)

Unknown directive type "attribute".

```
.. attribute:: st_atime_ns

   Time of most recent access expressed in nanoseconds as an integer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2732)

Unknown directive type "attribute".

```
.. attribute:: st_mtime_ns

   Time of most recent content modification expressed in nanoseconds as an
   integer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2737)

Unknown directive type "attribute".

```
.. attribute:: st_ctime_ns

   Platform dependent:

   * the time of most recent metadata change on Unix,
   * the time of creation on Windows, expressed in nanoseconds as an
     integer.
```

Note

The exact meaning and resolution of the :attr:`st_atime`, :attr:`st_mtime`, and :attr:`st_ctime` attributes depend on the operating system and the file system. For example, on Windows systems using the FAT or FAT32 file systems, :attr:`st_mtime` has 2-second resolution, and :attr:`st_atime` has only 1-day resolution. See your operating system documentation for details.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2747); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2747); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2747); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2747); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2747); *backlink*

Unknown interpreted text role "attr".

Similarly, although :attr:`st_atime_ns`, :attr:`st_mtime_ns`, and :attr:`st_ctime_ns` are always expressed in nanoseconds, many systems do not provide nanosecond precision. On systems that do provide nanosecond precision, the floating-point object used to store :attr:`st_atime`, :attr:`st_mtime`, and :attr:`st_ctime` cannot preserve all of it, and as such will be slightly inexact. If you need the exact timestamps you should always use :attr:`st_atime_ns`, :attr:`st_mtime_ns`, and :attr:`st_ctime_ns`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2754); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2754); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2754); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2754); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2754); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2754); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2754); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2754); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2754); *backlink*

Unknown interpreted text role "attr".

On some Unix systems (such as Linux), the following attributes may also be available:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2766)

Unknown directive type "attribute".

```
.. attribute:: st_blocks

   Number of 512-byte blocks allocated for file.
   This may be smaller than :attr:`st_size`/512 when the file has holes.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2771)

Unknown directive type "attribute".

```
.. attribute:: st_blksize

   "Preferred" blocksize for efficient file system I/O. Writing to a file in
   smaller chunks may cause an inefficient read-modify-rewrite.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2776)

Unknown directive type "attribute".

```
.. attribute:: st_rdev

   Type of device if an inode device.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2780)

Unknown directive type "attribute".

```
.. attribute:: st_flags

   User defined flags for file.
```

On other Unix systems (such as FreeBSD), the following attributes may be available (but may be only filled out if root tries to use them):

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2787)

Unknown directive type "attribute".

```
.. attribute:: st_gen

   File generation number.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2791)

Unknown directive type "attribute".

```
.. attribute:: st_birthtime

   Time of file creation.
```

On Solaris and derivatives, the following attributes may also be available:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2798)

Unknown directive type "attribute".

```
.. attribute:: st_fstype

   String that uniquely identifies the type of the filesystem that
   contains the file.
```

On macOS systems, the following attributes may also be available:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2805)

Unknown directive type "attribute".

```
.. attribute:: st_rsize

   Real size of the file.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2809)

Unknown directive type "attribute".

```
.. attribute:: st_creator

   Creator of the file.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2813)

Unknown directive type "attribute".

```
.. attribute:: st_type

   File type.
```

On Windows systems, the following attributes are also available:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2819)

Unknown directive type "attribute".

```
.. attribute:: st_file_attributes

   Windows file attributes: ``dwFileAttributes`` member of the
   ``BY_HANDLE_FILE_INFORMATION`` structure returned by
   :c:func:`GetFileInformationByHandle`. See the ``FILE_ATTRIBUTE_*``
   constants in the :mod:`stat` module.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2826)

Unknown directive type "attribute".

```
.. attribute:: st_reparse_tag

   When :attr:`st_file_attributes` has the ``FILE_ATTRIBUTE_REPARSE_POINT``
   set, this field contains the tag identifying the type of reparse point.
   See the ``IO_REPARSE_TAG_*`` constants in the :mod:`stat` module.
```

The standard module :mod:`stat` defines functions and constants that are useful for extracting information from a :c:type:`stat` structure. (On Windows, some items are filled with dummy values.)

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2832); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2832); *backlink*

Unknown interpreted text role "c:type".

For backward compatibility, a :class:`stat_result` instance is also accessible as a tuple of at least 10 integers giving the most important (and portable) members of the :c:type:`stat` structure, in the order :attr:`st_mode`, :attr:`st_ino`, :attr:`st_dev`, :attr:`st_nlink`, :attr:`st_uid`, :attr:`st_gid`, :attr:`st_size`, :attr:`st_atime`, :attr:`st_mtime`, :attr:`st_ctime`. More items may be added at the end by some implementations. For compatibility with older Python versions, accessing :class:`stat_result` as a tuple always returns integers.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "c:type".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2836); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2845)

Unknown directive type "versionadded".

```
.. versionadded:: 3.3
   Added the :attr:`st_atime_ns`, :attr:`st_mtime_ns`, and
   :attr:`st_ctime_ns` members.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2849)

Unknown directive type "versionadded".

```
.. versionadded:: 3.5
   Added the :attr:`st_file_attributes` member on Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2852)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.5
   Windows now returns the file index as :attr:`st_ino` when
   available.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2856)

Unknown directive type "versionadded".

```
.. versionadded:: 3.7
   Added the :attr:`st_fstype` member to Solaris/derivatives.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2859)

Unknown directive type "versionadded".

```
.. versionadded:: 3.8
   Added the :attr:`st_reparse_tag` member on Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2862)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.8
   On Windows, the :attr:`st_mode` member now identifies special
   files as :const:`S_IFCHR`, :const:`S_IFIFO` or :const:`S_IFBLK`
   as appropriate.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2867)

Unknown directive type "function".

```
.. function:: statvfs(path)

   Perform a :c:func:`statvfs` system call on the given path.  The return value is
   an object whose attributes describe the filesystem on the given path, and
   correspond to the members of the :c:type:`statvfs` structure, namely:
   :attr:`f_bsize`, :attr:`f_frsize`, :attr:`f_blocks`, :attr:`f_bfree`,
   :attr:`f_bavail`, :attr:`f_files`, :attr:`f_ffree`, :attr:`f_favail`,
   :attr:`f_flag`, :attr:`f_namemax`, :attr:`f_fsid`.

   Two module-level constants are defined for the :attr:`f_flag` attribute's
   bit-flags: if :const:`ST_RDONLY` is set, the filesystem is mounted
   read-only, and if :const:`ST_NOSUID` is set, the semantics of
   setuid/setgid bits are disabled or not supported.

   Additional module-level constants are defined for GNU/glibc based systems.
   These are :const:`ST_NODEV` (disallow access to device special files),
   :const:`ST_NOEXEC` (disallow program execution), :const:`ST_SYNCHRONOUS`
   (writes are synced at once), :const:`ST_MANDLOCK` (allow mandatory locks on an FS),
   :const:`ST_WRITE` (write on file/directory/symlink), :const:`ST_APPEND`
   (append-only file), :const:`ST_IMMUTABLE` (immutable file), :const:`ST_NOATIME`
   (do not update access times), :const:`ST_NODIRATIME` (do not update directory access
   times), :const:`ST_RELATIME` (update atime relative to mtime/ctime).

   This function can support :ref:`specifying a file descriptor <path_fd>`.

   .. availability:: Unix.

   .. versionchanged:: 3.2
      The :const:`ST_RDONLY` and :const:`ST_NOSUID` constants were added.

   .. versionadded:: 3.3
      Added support for specifying *path* as an open file descriptor.

   .. versionchanged:: 3.4
      The :const:`ST_NODEV`, :const:`ST_NOEXEC`, :const:`ST_SYNCHRONOUS`,
      :const:`ST_MANDLOCK`, :const:`ST_WRITE`, :const:`ST_APPEND`,
      :const:`ST_IMMUTABLE`, :const:`ST_NOATIME`, :const:`ST_NODIRATIME`,
      and :const:`ST_RELATIME` constants were added.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.

   .. versionadded:: 3.7
      Added :attr:`f_fsid`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 2913)

Unknown directive type "data".

```
.. data:: supports_dir_fd

   A :class:`set` object indicating which functions in the :mod:`os`
   module accept an open file descriptor for their *dir_fd* parameter.
   Different platforms provide different features, and the underlying
   functionality Python uses to implement the *dir_fd* parameter is not
   available on all platforms Python supports.  For consistency's sake,
   functions that may support *dir_fd* always allow specifying the
   parameter, but will throw an exception if the functionality is used
   when it's not locally available. (Specifying ``None`` for *dir_fd*
   is always supported on all platforms.)

   To check whether a particular function accepts an open file descriptor
   for its *dir_fd* parameter, use the ``in`` operator on ``supports_dir_fd``.
   As an example, this expression evaluates to ``True`` if :func:`os.stat`
   accepts open file descriptors for *dir_fd* on the local platform::

       os.stat in os.supports_dir_fd

   Currently *dir_fd* parameters only work on Unix platforms;
   none of them work on Windows.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2938)

Unknown directive type "data".

```
.. data:: supports_effective_ids

   A :class:`set` object indicating whether :func:`os.access` permits
   specifying ``True`` for its *effective_ids* parameter on the local platform.
   (Specifying ``False`` for *effective_ids* is always supported on all
   platforms.)  If the local platform supports it, the collection will contain
   :func:`os.access`; otherwise it will be empty.

   This expression evaluates to ``True`` if :func:`os.access` supports
   ``effective_ids=True`` on the local platform::

       os.access in os.supports_effective_ids

   Currently *effective_ids* is only supported on Unix platforms;
   it does not work on Windows.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2957)

Unknown directive type "data".

```
.. data:: supports_fd

   A :class:`set` object indicating which functions in the
   :mod:`os` module permit specifying their *path* parameter as an open file
   descriptor on the local platform.  Different platforms provide different
   features, and the underlying functionality Python uses to accept open file
   descriptors as *path* arguments is not available on all platforms Python
   supports.

   To determine whether a particular function permits specifying an open file
   descriptor for its *path* parameter, use the ``in`` operator on
   ``supports_fd``. As an example, this expression evaluates to ``True`` if
   :func:`os.chdir` accepts open file descriptors for *path* on your local
   platform::

       os.chdir in os.supports_fd

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 2977)

Unknown directive type "data".

```
.. data:: supports_follow_symlinks

   A :class:`set` object indicating which functions in the :mod:`os` module
   accept ``False`` for their *follow_symlinks* parameter on the local platform.
   Different platforms provide different features, and the underlying
   functionality Python uses to implement *follow_symlinks* is not available
   on all platforms Python supports.  For consistency's sake, functions that
   may support *follow_symlinks* always allow specifying the parameter, but
```

```
    will throw an exception if the functionality is used when it's not locally
    available.  (Specifying ``True`` for *follow_symlinks* is always supported
    on all platforms.)

    To check whether a particular function accepts ``False`` for its
    *follow_symlinks* parameter, use the ``in`` operator on
    ``supports_follow_symlinks``.  As an example, this expression evaluates
    to ``True`` if you may specify ``follow_symlinks=False`` when calling
    :func:`os.stat` on the local platform::

        os.stat in os.supports_follow_symlinks

    .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3000)

Unknown directive type "function".

```
.. function:: symlink(src, dst, target_is_directory=False, *, dir_fd=None)

   Create a symbolic link pointing to *src* named *dst*.

   On Windows, a symlink represents either a file or a directory, and does not
   morph to the target dynamically.  If the target is present, the type of the
   symlink will be created to match. Otherwise, the symlink will be created
   as a directory if *target_is_directory* is ``True`` or a file symlink (the
   default) otherwise.  On non-Windows platforms, *target_is_directory* is ignored.

   This function can support :ref:`paths relative to directory descriptors
   <dir_fd>`.

   .. note::

      On newer versions of Windows 10, unprivileged accounts can create symlinks
      if Developer Mode is enabled. When Developer Mode is not available/enabled,
      the *SeCreateSymbolicLinkPrivilege* privilege is required, or the process
      must be run as an administrator.


      :exc:`OSError` is raised when the function is called by an unprivileged
      user.

   .. audit-event:: os.symlink src,dst,dir_fd os.symlink

   .. availability:: Unix, Windows.

   .. versionchanged:: 3.2
      Added support for Windows 6.0 (Vista) symbolic links.

   .. versionadded:: 3.3
      Added the *dir_fd* argument, and now allow *target_is_directory*
      on non-Windows platforms.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object` for *src* and *dst*.

   .. versionchanged:: 3.8
      Added support for unelevated symlinks on Windows with Developer Mode.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3042)

Unknown directive type "function".

```
.. function:: sync()

   Force write of everything to disk.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3051)

Unknown directive type "function".

```
.. function:: truncate(path, length)
```

```
   Truncate the file corresponding to *path*, so that it is at most
   *length* bytes in size.

   This function can support :ref:`specifying a file descriptor <path_fd>`.

   .. audit-event:: os.truncate path,length os.truncate

   .. availability:: Unix, Windows.

   .. versionadded:: 3.3

   .. versionchanged:: 3.5
      Added support for Windows

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)os.rst`, line 3071)

Unknown directive type "function".

```
.. function:: unlink(path, *, dir_fd=None)

   Remove (delete) the file *path*.  This function is semantically
   identical to :func:`remove`; the ``unlink`` name is its
   traditional Unix name.  Please see the documentation for
   :func:`remove` for further information.

   .. audit-event:: os.remove path,dir_fd os.unlink

   .. versionadded:: 3.3
      The *dir_fd* parameter.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main)(Doc)(library)os.rst`, line 3087)

Unknown directive type "function".

```
.. function:: utime(path, times=None, *[, ns], dir_fd=None, follow_symlinks=True)

   Set the access and modified times of the file specified by *path*.

   :func:`utime` takes two optional parameters, *times* and *ns*.
   These specify the times set on *path* and are used as follows:

   - If *ns* is specified,
     it must be a 2-tuple of the form ``(atime_ns, mtime_ns)``
     where each member is an int expressing nanoseconds.
   - If *times* is not ``None``,
     it must be a 2-tuple of the form ``(atime, mtime)``
     where each member is an int or float expressing seconds.
   - If *times* is ``None`` and *ns* is unspecified,
     this is equivalent to specifying ``ns=(atime_ns, mtime_ns)``
     where both times are the current time.

   It is an error to specify tuples for both *times* and *ns*.

   Note that the exact times you set here may not be returned by a subsequent
   :func:`~os.stat` call, depending on the resolution with which your operating
   system records access and modification times; see :func:`~os.stat`. The best
   way to preserve exact times is to use the *st_atime_ns* and *st_mtime_ns*
   fields from the :func:`os.stat` result object with the *ns* parameter to
   `utime`.

   This function can support :ref:`specifying a file descriptor <path_fd>`,
   :ref:`paths relative to directory descriptors <dir_fd>` and :ref:`not
   following symlinks <follow_symlinks>`.

   .. audit-event:: os.utime path,times,ns,dir_fd os.utime

   .. versionadded:: 3.3
      Added support for specifying *path* as an open file descriptor,
      and the *dir_fd*, *follow_symlinks*, and *ns* parameters.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3127)

Unknown directive type "function".

```
.. function:: walk(top, topdown=True, onerror=None, followlinks=False)

   .. index::
      single: directory; walking
      single: directory; traversal

   Generate the file names in a directory tree by walking the tree
   either top-down or bottom-up. For each directory in the tree rooted at directory
   *top* (including *top* itself), it yields a 3-tuple ``(dirpath, dirnames,
   filenames)``.

   *dirpath* is a string, the path to the directory.  *dirnames* is a list of the
   names of the subdirectories in *dirpath* (excluding ``'.'`` and ``'..'``).
   *filenames* is a list of the names of the non-directory files in *dirpath*.
   Note that the names in the lists contain no path components.  To get a full path
   (which begins with *top*) to a file or directory in *dirpath*, do
   ``os.path.join(dirpath, name)``.  Whether or not the lists are sorted
   depends on the file system.  If a file is removed from or added to the
   *dirpath* directory during generating the lists, whether a name for that
   file be included is unspecified.

   If optional argument *topdown* is ``True`` or not specified, the triple for a
   directory is generated before the triples for any of its subdirectories
   (directories are generated top-down).  If *topdown* is ``False``, the triple
   for a directory is generated after the triples for all of its subdirectories
   (directories are generated bottom-up). No matter the value of *topdown*, the
   list of subdirectories is retrieved before the tuples for the directory and
   its subdirectories are generated.

   When *topdown* is ``True``, the caller can modify the *dirnames* list in-place
   (perhaps using :keyword:`del` or slice assignment), and :func:`walk` will only
   recurse into the subdirectories whose names remain in *dirnames*; this can be
   used to prune the search, impose a specific order of visiting, or even to inform
   :func:`walk` about directories the caller creates or renames before it resumes
   :func:`walk` again.  Modifying *dirnames* when *topdown* is ``False`` has
   no effect on the behavior of the walk, because in bottom-up mode the directories
   in *dirnames* are generated before *dirpath* itself is generated.

   By default, errors from the :func:`scandir` call are ignored.  If optional
   argument *onerror* is specified, it should be a function; it will be called with
   one argument, an :exc:`OSError` instance.  It can report the error to continue
   with the walk, or raise the exception to abort the walk.  Note that the filename
   is available as the ``filename`` attribute of the exception object.

   By default, :func:`walk` will not walk down into symbolic links that resolve to
   directories. Set *followlinks* to ``True`` to visit directories pointed to by
   symlinks, on systems that support them.

   .. note::

      Be aware that setting *followlinks* to ``True`` can lead to infinite
      recursion if a link points to a parent directory of itself. :func:`walk`
      does not keep track of the directories it visited already.

   .. note::

      If you pass a relative pathname, don't change the current working directory
      between resumptions of :func:`walk`.  :func:`walk` never changes the current
      directory, and assumes that its caller doesn't either.

   This example displays the number of bytes taken by non-directory files in each
   directory under the starting directory, except that it doesn't look under any
   CVS subdirectory::

      import os
      from os.path import join, getsize
      for root, dirs, files in os.walk('python/Lib/email'):
          print(root, "consumes", end=" ")
          print(sum(getsize(join(root, name)) for name in files), end=" ")
          print("bytes in", len(files), "non-directory files")
          if 'CVS' in dirs:
              dirs.remove('CVS')  # don't visit CVS directories

   In the next example (simple implementation of :func:`shutil.rmtree`),
   walking the tree bottom-up is essential, :func:`rmdir` doesn't allow
   deleting a directory before the directory is empty::

      # Delete everything reachable from the directory named in "top",
      # assuming there are no symbolic links.
      # CAUTION:  This is dangerous!  For example, if top == '/', it
      # could delete all your disk files.
      import os
      for root, dirs, files in os.walk(top, topdown=False):
          for name in files:
              os.remove(os.path.join(root, name))
          for name in dirs:
              os.rmdir(os.path.join(root, name))

   .. audit-event:: os.walk top,topdown,onerror,followlinks os.walk

   .. versionchanged:: 3.5
      This function now calls :func:`os.scandir` instead of :func:`os.listdir`,
      making it faster by reducing the number of calls to :func:`os.stat`.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3225)

Unknown directive type "function".

```
.. function:: fwalk(top='.', topdown=True, onerror=None, *, follow_symlinks=False, dir_fd=None)

   .. index::
      single: directory; walking
      single: directory; traversal

   This behaves exactly like :func:`walk`, except that it yields a 4-tuple
   ``(dirpath, dirnames, filenames, dirfd)``, and it supports ``dir_fd``.

   *dirpath*, *dirnames* and *filenames* are identical to :func:`walk` output,
   and *dirfd* is a file descriptor referring to the directory *dirpath*.

   This function always supports :ref:`paths relative to directory descriptors
   <dir_fd>` and :ref:`not following symlinks <follow_symlinks>`.  Note however
   that, unlike other functions, the :func:`fwalk` default value for
   *follow_symlinks* is ``False``.

   .. note::

      Since :func:`fwalk` yields file descriptors, those are only valid until
      the next iteration step, so you should duplicate them (e.g. with
      :func:`dup`) if you want to keep them longer.

   This example displays the number of bytes taken by non-directory files in each
   directory under the starting directory, except that it doesn't look under any
   CVS subdirectory::

      import os
      for root, dirs, files, rootfd in os.fwalk('python/Lib/email'):
          print(root, "consumes", end="")
          print(sum([os.stat(name, dir_fd=rootfd).st_size for name in files]),
                end="")
          print("bytes in", len(files), "non-directory files")
          if 'CVS' in dirs:
              dirs.remove('CVS')  # don't visit CVS directories

   In the next example, walking the tree bottom-up is essential:
   :func:`rmdir` doesn't allow deleting a directory before the directory is
   empty::

      # Delete everything reachable from the directory named in "top",
      # assuming there are no symbolic links.
      # CAUTION:  This is dangerous!  For example, if top == '/', it
      # could delete all your disk files.
      import os
      for root, dirs, files, rootfd in os.fwalk(top, topdown=False):
          for name in files:
              os.unlink(name, dir_fd=rootfd)
          for name in dirs:
              os.rmdir(name, dir_fd=rootfd)

   .. audit-event:: os.fwalk top,topdown,onerror,follow_symlinks,dir_fd os.fwalk

   .. availability:: Unix.

   .. versionadded:: 3.3

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.

   .. versionchanged:: 3.7
      Added support for :class:`bytes` paths.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3289)

Unknown directive type "function".

```
.. function:: memfd_create(name[, flags=os.MFD_CLOEXEC])

   Create an anonymous file and return a file descriptor that refers to it.
   *flags* must be one of the ``os.MFD_*`` constants available on the system
   (or a bitwise ORed combination of them).  By default, the new file
   descriptor is :ref:`non-inheritable <fd_inheritance>`.

   The name supplied in *name* is used as a filename and will be displayed as
   the target of the corresponding symbolic link in the directory
   ``/proc/self/fd/``. The displayed name is always prefixed with ``memfd:``
   and serves only for debugging purposes. Names do not affect the behavior of
   the file descriptor, and as such multiple files can have the same name
   without any side effects.

   .. availability:: Linux 3.17 or newer with glibc 2.27 or newer.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3308)

Unknown directive type "data".

```
.. data:: MFD_CLOEXEC
          MFD_ALLOW_SEALING
          MFD_HUGETLB
          MFD_HUGE_SHIFT
          MFD_HUGE_MASK
          MFD_HUGE_64KB
          MFD_HUGE_512KB
          MFD_HUGE_1MB
          MFD_HUGE_2MB
          MFD_HUGE_8MB
          MFD_HUGE_16MB
          MFD_HUGE_32MB
          MFD_HUGE_256MB
          MFD_HUGE_512MB
          MFD_HUGE_1GB
          MFD_HUGE_2GB
          MFD_HUGE_16GB

   These flags can be passed to :func:`memfd_create`.

   .. availability:: Linux 3.17 or newer with glibc 2.27 or newer.  The
      ``MFD_HUGE*`` flags are only available since Linux 4.14.

   .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3334)

Unknown directive type "function".

```
.. function:: eventfd(initval[, flags=os.EFD_CLOEXEC])

   Create and return an event file descriptor. The file descriptors supports
   raw :func:`read` and :func:`write` with a buffer size of 8,
   :func:`~select.select`, :func:`~select.poll` and similar. See man page
   :manpage:`eventfd(2)` for more information.  By default, the
   new file descriptor is :ref:`non-inheritable <fd_inheritance>`.

   *initval* is the initial value of the event counter. The initial value
   must be an 32 bit unsigned integer. Please note that the initial value is
   limited to a 32 bit unsigned int although the event counter is an unsigned
   64 bit integer with a maximum value of 2\ :sup:`64`\ -\ 2.

   *flags* can be constructed from :const:`EFD_CLOEXEC`,
   :const:`EFD_NONBLOCK`, and :const:`EFD_SEMAPHORE`.

   If :const:`EFD_SEMAPHORE` is specified and the event counter is non-zero,
   :func:`eventfd_read` returns 1 and decrements the counter by one.

   If :const:`EFD_SEMAPHORE` is not specified and the event counter is
   non-zero, :func:`eventfd_read` returns the current event counter value and
   resets the counter to zero.

   If the event counter is zero and :const:`EFD_NONBLOCK` is not
   specified, :func:`eventfd_read` blocks.
   :func:`eventfd_write` increments the event counter. Write blocks if the
   write operation would increment the counter to a value larger than
   2\ :sup:`64`\ -\ 2.

   Example::

       import os

       # semaphore with start value '1'
       fd = os.eventfd(1, os.EFD_SEMAPHORE | os.EFC_CLOEXEC)
       try:
           # acquire semaphore
           v = os.eventfd_read(fd)
           try:
               do_work()
           finally:
               # release semaphore
               os.eventfd_write(fd, v)
       finally:
           os.close(fd)

   .. availability:: Linux 2.6.27 or newer with glibc 2.8 or newer.

   .. versionadded:: 3.10
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3385)

Unknown directive type "function".

```
.. function:: eventfd_read(fd)

   Read value from an :func:`eventfd` file descriptor and return a 64 bit
   unsigned int. The function does not verify that *fd* is an :func:`eventfd`.

   .. availability:: See :func:`eventfd`

   .. versionadded:: 3.10
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3394)

Unknown directive type "function".

```
.. function:: eventfd_write(fd, value)

   Add value to an :func:`eventfd` file descriptor. *value* must be a 64 bit
   unsigned int. The function does not verify that *fd* is an :func:`eventfd`.

   .. availability:: See :func:`eventfd`

   .. versionadded:: 3.10
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3403)

Unknown directive type "data".

```
.. data:: EFD_CLOEXEC

   Set close-on-exec flag for new :func:`eventfd` file descriptor.

   .. availability:: See :func:`eventfd`

   .. versionadded:: 3.10
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3411)

Unknown directive type "data".

```
.. data:: EFD_NONBLOCK

   Set :const:`O_NONBLOCK` status flag for new :func:`eventfd` file
   descriptor.

   .. availability:: See :func:`eventfd`

   .. versionadded:: 3.10
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3420)

Unknown directive type "data".

```
.. data:: EFD_SEMAPHORE

   Provide semaphore-like semantics for reads from a :func:`eventfd` file
   descriptor. On read the internal counter is decremented by one.

   .. availability:: Linux 2.6.30 or newer with glibc 2.8 or newer.

   .. versionadded:: 3.10
```

Linux extended attributes

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3433)

Unknown directive type "versionadded".

```
.. versionadded:: 3.3
```

These functions are all available on Linux only.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3437)

Unknown directive type "function".

```
.. function:: getxattr(path, attribute, *, follow_symlinks=True)

   Return the value of the extended filesystem attribute *attribute* for
   *path*. *attribute* can be bytes or str (directly or indirectly through the
   :class:`PathLike` interface). If it is str, it is encoded with the filesystem
   encoding.

   This function can support :ref:`specifying a file descriptor <path_fd>` and
   :ref:`not following symlinks <follow_symlinks>`.

   .. audit-event:: os.getxattr path,attribute os.getxattr

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object` for *path* and *attribute*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3453)

Unknown directive type "function".

```
.. function:: listxattr(path=None, *, follow_symlinks=True)

   Return a list of the extended filesystem attributes on *path*.  The
   attributes in the list are represented as strings decoded with the filesystem
   encoding.  If *path* is ``None``, :func:`listxattr` will examine the current
   directory.

   This function can support :ref:`specifying a file descriptor <path_fd>` and
   :ref:`not following symlinks <follow_symlinks>`.

   .. audit-event:: os.listxattr path os.listxattr

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3469)

Unknown directive type "function".

```
.. function:: removexattr(path, attribute, *, follow_symlinks=True)

   Removes the extended filesystem attribute *attribute* from *path*.
   *attribute* should be bytes or str (directly or indirectly through the
   :class:`PathLike` interface). If it is a string, it is encoded
   with the :term:`filesystem encoding and error handler`.
   This function can support :ref:`specifying a file descriptor <path_fd>` and
   :ref:`not following symlinks <follow_symlinks>`.

   .. audit-event:: os.removexattr path,attribute os.removexattr

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object` for *path* and *attribute*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3485)

Unknown directive type "function".

```
.. function:: setxattr(path, attribute, value, flags=0, *, follow_symlinks=True)

   Set the extended filesystem attribute *attribute* on *path* to *value*.
   *attribute* must be a bytes or str with no embedded NULs (directly or
   indirectly through the :class:`PathLike` interface). If it is a str,
   it is encoded with the :term:`filesystem encoding and error handler`.  *flags* may be
   :data:`XATTR_REPLACE` or :data:`XATTR_CREATE`. If :data:`XATTR_REPLACE` is
   given and the attribute does not exist, ``ENODATA`` will be raised.
   If :data:`XATTR_CREATE` is given and the attribute already exists, the
   attribute will not be created and ``EEXISTS`` will be raised.

   This function can support :ref:`specifying a file descriptor <path_fd>` and
   :ref:`not following symlinks <follow_symlinks>`.

   .. note::

      A bug in Linux kernel versions less than 2.6.39 caused the flags argument
      to be ignored on some filesystems.

   .. audit-event:: os.setxattr path,attribute,value,flags os.setxattr

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object` for *path* and *attribute*.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3510)

Unknown directive type "data".

```
.. data:: XATTR_SIZE_MAX

   The maximum size the value of an extended attribute can be. Currently, this
   is 64 KiB on Linux.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3516)

Unknown directive type "data".

```
.. data:: XATTR_CREATE

   This is a possible value for the flags argument in :func:`setxattr`. It
   indicates the operation must create an attribute.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3522)

Unknown directive type "data".

```
.. data:: XATTR_REPLACE

   This is a possible value for the flags argument in :func:`setxattr`. It
   indicates the operation must replace an existing attribute.
```

Process Management

These functions may be used to create and manage processes.

The various :func:`exec\* <execl>` functions take a list of arguments for the new program loaded into the process. In each case, the first of these arguments is passed to the new program as its own name rather than as an argument a user may have typed on a command line. For the C programmer, this is the `argv[0]` passed to a program's :c:func:`main`. For example, `os.execv('/bin/echo', ['foo', 'bar'])` will only print `bar` on standard output; `foo` will seem to be ignored.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3535); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3535); *backlink*

Unknown interpreted text role "c:func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3544)

Unknown directive type "function".

```
.. function:: abort()

   Generate a :const:`SIGABRT` signal to the current process.  On Unix, the default
   behavior is to produce a core dump; on Windows, the process immediately returns
   an exit code of ``3``.  Be aware that calling this function will not call the
   Python signal handler registered for :const:`SIGABRT` with
   :func:`signal.signal`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3553)

Unknown directive type "function".

```
.. function:: add_dll_directory(path)

   Add a path to the DLL search path.

   This search path is used when resolving dependencies for imported
   extension modules (the module itself is resolved through sys.path),
   and also by :mod:`ctypes`.

   Remove the directory by calling **close()** on the returned object
   or using it in a :keyword:`with` statement.

   See the `Microsoft documentation
   <https://msdn.microsoft.com/44228cf2-6306-466c-8f16-f513cd3ba8b5>`_
   for more information about how DLLs are loaded.

   .. audit-event:: os.add_dll_directory path os.add_dll_directory

   .. availability:: Windows.

   .. versionadded:: 3.8
      Previous versions of CPython would resolve DLLs using the default
      behavior for the current process. This led to inconsistencies,
      such as only sometimes searching :envvar:`PATH` or the current
      working directory, and OS functions such as ``AddDllDirectory``
      having no effect.

      In 3.8, the two primary ways DLLs are loaded now explicitly
      override the process-wide behavior to ensure consistency. See the
      :ref:`porting notes <bpo-36085-whatsnew>` for information on
      updating libraries.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3585)

Unknown directive type "function".

```
.. function:: execl(path, arg0, arg1, ...)
              execle(path, arg0, arg1, ..., env)
              execlp(file, arg0, arg1, ...)
              execlpe(file, arg0, arg1, ..., env)
              execv(path, args)
              execve(path, args, env)
              execvp(file, args)
              execvpe(file, args, env)

   These functions all execute a new program, replacing the current process; they
   do not return.  On Unix, the new executable is loaded into the current process,
   and will have the same process id as the caller.  Errors will be reported as
   :exc:`OSError` exceptions.

   The current process is replaced immediately. Open file objects and
   descriptors are not flushed, so if there may be data buffered
   on these open files, you should flush them using
   :func:`sys.stdout.flush` or :func:`os.fsync` before calling an
   :func:`exec\* <execl>` function.

   The "l" and "v" variants of the :func:`exec\* <execl>` functions differ in how
   command-line arguments are passed.  The "l" variants are perhaps the easiest
   to work with if the number of parameters is fixed when the code is written; the
   individual parameters simply become additional parameters to the :func:`execl\*`
   functions.  The "v" variants are good when the number of parameters is
   variable, with the arguments being passed in a list or tuple as the *args*
   parameter.  In either case, the arguments to the child process should start with
   the name of the command being run, but this is not enforced.

   The variants which include a "p" near the end (:func:`execlp`,
   :func:`execlpe`, :func:`execvp`, and :func:`execvpe`) will use the
   :envvar:`PATH` environment variable to locate the program *file*.  When the
   environment is being replaced (using one of the :func:`exec\*e <execl>` variants,
   discussed in the next paragraph), the new environment is used as the source of
   the :envvar:`PATH` variable. The other variants, :func:`execl`, :func:`execle`,
   :func:`execv`, and :func:`execve`, will not use the :envvar:`PATH` variable to
   locate the executable; *path* must contain an appropriate absolute or relative
   path.

   For :func:`execle`, :func:`execlpe`, :func:`execve`, and :func:`execvpe` (note
   that these all end in "e"), the *env* parameter must be a mapping which is
   used to define the environment variables for the new process (these are used
   instead of the current process' environment); the functions :func:`execl`,
   :func:`execlp`, :func:`execv`, and :func:`execvp` all cause the new process to
   inherit the environment of the current process.

   For :func:`execve` on some platforms, *path* may also be specified as an open
   file descriptor.  This functionality may not be supported on your platform;
   you can check whether or not it is available using :data:`os.supports_fd`.
   If it is unavailable, using it will raise a :exc:`NotImplementedError`.

   .. audit-event:: os.exec path,args,env os.execl

   .. availability:: Unix, Windows.

   .. versionadded:: 3.3
      Added support for specifying *path* as an open file descriptor
      for :func:`execve`.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3647)

Unknown directive type "function".

```
.. function:: _exit(n)

   Exit the process with status *n*, without calling cleanup handlers, flushing
   stdio buffers, etc.

   .. note::

      The standard way to exit is ``sys.exit(n)``.  :func:`_exit` should
      normally only be used in the child process after a :func:`fork`.
```

The following exit codes are defined and can be used with :func:`_exit`, although they are not required. These are typically used for system programs written in Python, such as a mail server's external command delivery program.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3657); *backlink*

Unknown interpreted text role "func".

Note

Some of these may not be available on all Unix platforms, since there is some variation. These constants are defined where they are defined by the underlying platform.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3668)

Unknown directive type "data".

```
.. data:: EX_OK

   Exit code that means no error occurred. May be taken from the defined value of
   ``EXIT_SUCCESS`` on some platforms. Generally has a value of zero.

   .. availability:: Unix, Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3676)

Unknown directive type "data".

```
.. data:: EX_USAGE

   Exit code that means the command was used incorrectly, such as when the wrong
   number of arguments are given.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3684)

Unknown directive type "data".

```
.. data:: EX_DATAERR

   Exit code that means the input data was incorrect.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3691)

Unknown directive type "data".

```
.. data:: EX_NOINPUT

   Exit code that means an input file did not exist or was not readable.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3698)

Unknown directive type "data".

```
.. data:: EX_NOUSER

   Exit code that means a specified user did not exist.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3705)

Unknown directive type "data".

```
.. data:: EX_NOHOST

   Exit code that means a specified host did not exist.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3712)

Unknown directive type "data".

```
.. data:: EX_UNAVAILABLE
```

```
   Exit code that means that a required service is unavailable.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3719)

Unknown directive type "data".

```
.. data:: EX_SOFTWARE

   Exit code that means an internal software error was detected.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3726)

Unknown directive type "data".

```
.. data:: EX_OSERR

   Exit code that means an operating system error was detected, such as the
   inability to fork or create a pipe.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3734)

Unknown directive type "data".

```
.. data:: EX_OSFILE

   Exit code that means some system file did not exist, could not be opened, or had
   some other kind of error.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3742)

Unknown directive type "data".

```
.. data:: EX_CANTCREAT

   Exit code that means a user specified output file could not be created.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3749)

Unknown directive type "data".

```
.. data:: EX_IOERR

   Exit code that means that an error occurred while doing I/O on some file.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 3756)

Unknown directive type "data".

```
.. data:: EX_TEMPFAIL

   Exit code that means a temporary failure occurred.  This indicates something
   that may not really be an error, such as a network connection that couldn't be
   made during a retryable operation.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)os.rst, line 3765)

Unknown directive type "data".

```
.. data:: EX_PROTOCOL

   Exit code that means that a protocol exchange was illegal, invalid, or not
   understood.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)os.rst, line 3773)

Unknown directive type "data".

```
.. data:: EX_NOPERM

   Exit code that means that there were insufficient permissions to perform the
   operation (but not intended for file system problems).

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)os.rst, line 3781)

Unknown directive type "data".

```
.. data:: EX_CONFIG

   Exit code that means that some kind of configuration error occurred.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)os.rst, line 3788)

Unknown directive type "data".

```
.. data:: EX_NOTFOUND

   Exit code that means something like "an entry was not found".

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\ (cpython-main) (Doc) (library)os.rst, line 3795)

Unknown directive type "function".

```
.. function:: fork()

   Fork a child process.  Return ``0`` in the child and the child's process id in the
   parent.  If an error occurs :exc:`OSError` is raised.

   Note that some platforms including FreeBSD <= 6.3 and Cygwin have
   known issues when using ``fork()`` from a thread.

   .. audit-event:: os.fork "" os.fork

   .. versionchanged:: 3.8
      Calling ``fork()`` in a subinterpreter is no longer supported
      (:exc:`RuntimeError` is raised).

   .. warning::

      See :mod:`ssl` for applications that use the SSL module with fork().

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3816)

Unknown directive type "function".

```
.. function:: forkpty()

   Fork a child process, using a new pseudo-terminal as the child's controlling
   terminal. Return a pair of ``(pid, fd)``, where *pid* is ``0`` in the child, the
   new child's process id in the parent, and *fd* is the file descriptor of the
   master end of the pseudo-terminal.  For a more portable approach, use the
   :mod:`pty` module.  If an error occurs :exc:`OSError` is raised.

   .. audit-event:: os.forkpty "" os.forkpty

   .. versionchanged:: 3.8
      Calling ``forkpty()`` in a subinterpreter is no longer supported
      (:exc:`RuntimeError` is raised).

   .. availability:: some flavors of Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3833)

Unknown directive type "function".

```
.. function:: kill(pid, sig)

   .. index::
      single: process; killing
      single: process; signalling

   Send signal *sig* to the process *pid*.  Constants for the specific signals
   available on the host platform are defined in the :mod:`signal` module.

   Windows: The :data:`signal.CTRL_C_EVENT` and
   :data:`signal.CTRL_BREAK_EVENT` signals are special signals which can
   only be sent to console processes which share a common console window,
   e.g., some subprocesses. Any other value for *sig* will cause the process
   to be unconditionally killed by the TerminateProcess API, and the exit code
   will be set to *sig*. The Windows version of :func:`kill` additionally takes
   process handles to be killed.

   See also :func:`signal.pthread_kill`.

   .. audit-event:: os.kill pid,sig os.kill

   .. versionadded:: 3.2
      Windows support.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3858)

Unknown directive type "function".

```
.. function:: killpg(pgid, sig)

   .. index::
      single: process; killing
      single: process; signalling

   Send the signal *sig* to the process group *pgid*.

   .. audit-event:: os.killpg pgid,sig os.killpg

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3871)

Unknown directive type "function".

```
.. function:: nice(increment)

   Add *increment* to the process's "niceness".  Return the new niceness.
   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3878)

Unknown directive type "function".

```
.. function:: pidfd_open(pid, flags=0)

   Return a file descriptor referring to the process *pid*.  This descriptor can
   be used to perform process management without races and signals.  The *flags*
   argument is provided for future extensions; no flag values are currently
   defined.

   See the :manpage:`pidfd_open(2)` man page for more details.

   .. availability:: Linux 5.3+
   .. versionadded:: 3.9
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3891)

Unknown directive type "function".

```
.. function:: plock(op)

   Lock program segments into memory.  The value of *op* (defined in
   ``<sys/lock.h>``) determines which segments are locked.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3899)

Unknown directive type "function".

```
.. function:: popen(cmd, mode='r', buffering=-1)

   Open a pipe to or from command *cmd*.
   The return value is an open file object
   connected to the pipe, which can be read or written depending on whether *mode*
   is ``'r'`` (default) or ``'w'``. The *buffering* argument has the same meaning as
   the corresponding argument to the built-in :func:`open` function. The
   returned file object reads or writes text strings rather than bytes.

   The ``close`` method returns :const:`None` if the subprocess exited
   successfully, or the subprocess's return code if there was an
   error. On POSIX systems, if the return code is positive it
   represents the return value of the process left-shifted by one
   byte.  If the return code is negative, the process was terminated
   by the signal given by the negated value of the return code.  (For
   example, the return value might be ``- signal.SIGKILL`` if the
   subprocess was killed.)  On Windows systems, the return value
   contains the signed integer return code from the child process.

   On Unix, :func:`waitstatus_to_exitcode` can be used to convert the ``close``
   method result (exit status) into an exit code if it is not ``None``. On
   Windows, the ``close`` method result is directly the exit code
   (or ``None``).

   This is implemented using :class:`subprocess.Popen`; see that class's
   documentation for more powerful ways to manage and communicate with
   subprocesses.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 3928)

Unknown directive type "function".

```
.. function:: posix_spawn(path, argv, env, *, file_actions=None, \
                          setpgroup=None, resetids=False, setsid=False, setsigmask=(), \
                          setsigdef=(), scheduler=None)

   Wraps the :c:func:`posix_spawn` C library API for use from Python.

   Most users should use :func:`subprocess.run` instead of :func:`posix_spawn`.
   The positional-only arguments *path*, *args*, and *env* are similar to
   :func:`execve`.

   The *path* parameter is the path to the executable file.  The *path* should
   contain a directory.  Use :func:`posix_spawnp` to pass an executable file
   without directory.

   The *file_actions* argument may be a sequence of tuples describing actions
   to take on specific file descriptors in the child process between the C
   library implementation's :c:func:`fork` and :c:func:`exec` steps.
   The first item in each tuple must be one of the three type indicator
   listed below describing the remaining tuple elements:

   .. data:: POSIX_SPAWN_OPEN

      (``os.POSIX_SPAWN_OPEN``, *fd*, *path*, *flags*, *mode*)

      Performs ``os.dup2(os.open(path, flags, mode), fd)``.

   .. data:: POSIX_SPAWN_CLOSE

      (``os.POSIX_SPAWN_CLOSE``, *fd*)

      Performs ``os.close(fd)``.

   .. data:: POSIX_SPAWN_DUP2

      (``os.POSIX_SPAWN_DUP2``, *fd*, *new_fd*)

      Performs ``os.dup2(fd, new_fd)``.

   These tuples correspond to the C library
   :c:func:`posix_spawn_file_actions_addopen`,
   :c:func:`posix_spawn_file_actions_addclose`, and
   :c:func:`posix_spawn_file_actions_adddup2` API calls used to prepare
   for the :c:func:`posix_spawn` call itself.

   The *setpgroup* argument will set the process group of the child to the value
   specified. If the value specified is 0, the child's process group ID will be
   made the same as its process ID. If the value of *setpgroup* is not set, the
   child will inherit the parent's process group ID. This argument corresponds
   to the C library :c:data:`POSIX_SPAWN_SETPGROUP` flag.

   If the *resetids* argument is ``True`` it will reset the effective UID and
   GID of the child to the real UID and GID of the parent process. If the
   argument is ``False``, then the child retains the effective UID and GID of
   the parent. In either case, if the set-user-ID and set-group-ID permission
   bits are enabled on the executable file, their effect will override the
   setting of the effective UID and GID. This argument corresponds to the C
   library :c:data:`POSIX_SPAWN_RESETIDS` flag.

   If the *setsid* argument is ``True``, it will create a new session ID
   for `posix_spawn`. *setsid* requires :c:data:`POSIX_SPAWN_SETSID`
   or :c:data:`POSIX_SPAWN_SETSID_NP` flag. Otherwise, :exc:`NotImplementedError`
   is raised.

   The *setsigmask* argument will set the signal mask to the signal set
   specified. If the parameter is not used, then the child inherits the
   parent's signal mask. This argument corresponds to the C library
   :c:data:`POSIX_SPAWN_SETSIGMASK` flag.

   The *sigdef* argument will reset the disposition of all signals in the set
   specified. This argument corresponds to the C library
   :c:data:`POSIX_SPAWN_SETSIGDEF` flag.

   The *scheduler* argument must be a tuple containing the (optional) scheduler
   policy and an instance of :class:`sched_param` with the scheduler parameters.
   A value of ``None`` in the place of the scheduler policy indicates that is
   not being provided. This argument is a combination of the C library
   :c:data:`POSIX_SPAWN_SETSCHEDPARAM` and :c:data:`POSIX_SPAWN_SETSCHEDULER`
   flags.

   .. audit-event:: os.posix_spawn path,argv,env os.posix_spawn

   .. versionadded:: 3.8

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4014)

Unknown directive type "function".

```
.. function:: posix_spawnp(path, argv, env, *, file_actions=None, \
                           setpgroup=None, resetids=False, setsid=False, setsigmask=(), \
                           setsigdef=(), scheduler=None)
```

```
   Wraps the :c:func:`posix_spawnp` C library API for use from Python.

   Similar to :func:`posix_spawn` except that the system searches
   for the *executable* file in the list of directories specified by the
   :envvar:`PATH` environment variable (in the same way as for ``execvp(3)``).

   .. audit-event:: os.posix_spawn path,argv,env os.posix_spawnp

   .. versionadded:: 3.8

   .. availability:: See :func:`posix_spawn` documentation.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4031)

Unknown directive type "function".

```
.. function:: register_at_fork(*, before=None, after_in_parent=None, \
                               after_in_child=None)

   Register callables to be executed when a new child process is forked
   using :func:`os.fork` or similar process cloning APIs.
   The parameters are optional and keyword-only.
   Each specifies a different call point.

   * *before* is a function called before forking a child process.
   * *after_in_parent* is a function called from the parent process
     after forking a child process.
   * *after_in_child* is a function called from the child process.

   These calls are only made if control is expected to return to the
   Python interpreter.  A typical :mod:`subprocess` launch will not
   trigger them as the child is not going to re-enter the interpreter.

   Functions registered for execution before forking are called in
   reverse registration order.  Functions registered for execution
   after forking (either in the parent or in the child) are called
   in registration order.

   Note that :c:func:`fork` calls made by third-party C code may not
   call those functions, unless it explicitly calls :c:func:`PyOS_BeforeFork`,
   :c:func:`PyOS_AfterFork_Parent` and :c:func:`PyOS_AfterFork_Child`.

   There is no way to unregister a function.

   .. availability:: Unix.

   .. versionadded:: 3.7
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4064)

Unknown directive type "function".

```
.. function:: spawnl(mode, path, ...)
              spawnle(mode, path, ..., env)
              spawnlp(mode, file, ...)
              spawnlpe(mode, file, ..., env)
              spawnv(mode, path, args)
              spawnve(mode, path, args, env)
              spawnvp(mode, file, args)
              spawnvpe(mode, file, args, env)

   Execute the program *path* in a new process.

   (Note that the :mod:`subprocess` module provides more powerful facilities for
   spawning new processes and retrieving their results; using that module is
   preferable to using these functions.  Check especially the
   :ref:`subprocess-replacements` section.)

   If *mode* is :const:`P_NOWAIT`, this function returns the process id of the new
   process; if *mode* is :const:`P_WAIT`, returns the process's exit code if it
   exits normally, or ``-signal``, where *signal* is the signal that killed the
   process.  On Windows, the process id will actually be the process handle, so can
   be used with the :func:`waitpid` function.

   Note on VxWorks, this function doesn't return ``-signal`` when the new process is
   killed. Instead it raises OSError exception.

   The "l" and "v" variants of the :func:`spawn\* <spawnl>` functions differ in how
   command-line arguments are passed.  The "l" variants are perhaps the easiest
   to work with if the number of parameters is fixed when the code is written; the
   individual parameters simply become additional parameters to the
   :func:`spawnl\*` functions.  The "v" variants are good when the number of
   parameters is variable, with the arguments being passed in a list or tuple as
   the *args* parameter.  In either case, the arguments to the child process must
   start with the name of the command being run.

   The variants which include a second "p" near the end (:func:`spawnlp`,
   :func:`spawnlpe`, :func:`spawnvp`, and :func:`spawnvpe`) will use the
   :envvar:`PATH` environment variable to locate the program *file*.  When the
   environment is being replaced (using one of the :func:`spawn\*e <spawnl>` variants,
   discussed in the next paragraph), the new environment is used as the source of
   the :envvar:`PATH` variable.  The other variants, :func:`spawnl`,
   :func:`spawnle`, :func:`spawnv`, and :func:`spawnve`, will not use the
   :envvar:`PATH` variable to locate the executable; *path* must contain an
   appropriate absolute or relative path.

   For :func:`spawnle`, :func:`spawnlpe`, :func:`spawnve`, and :func:`spawnvpe`
   (note that these all end in "e"), the *env* parameter must be a mapping
   which is used to define the environment variables for the new process (they are
   used instead of the current process' environment); the functions
   :func:`spawnl`, :func:`spawnlp`, :func:`spawnv`, and :func:`spawnvp` all cause
   the new process to inherit the environment of the current process.  Note that
   keys and values in the *env* dictionary must be strings; invalid keys or
   values will cause the function to fail, with a return value of ``127``.

   As an example, the following calls to :func:`spawnlp` and :func:`spawnvpe` are
   equivalent::

      import os
      os.spawnlp(os.P_WAIT, 'cp', 'cp', 'index.html', '/dev/null')

      L = ['cp', 'index.html', '/dev/null']
      os.spawnvpe(os.P_WAIT, 'cp', L, os.environ)

   .. audit-event:: os.spawn mode,path,args,env os.spawnl

   .. availability:: Unix, Windows.  :func:`spawnlp`, :func:`spawnlpe`, :func:`spawnvp`
      and :func:`spawnvpe` are not available on Windows.  :func:`spawnle` and
      :func:`spawnve` are not thread-safe on Windows; we advise you to use the
      :mod:`subprocess` module instead.

   .. versionchanged:: 3.6
      Accepts a :term:`path-like object`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4137)

Unknown directive type "data".

```
.. data:: P_NOWAIT
          P_NOWAITO

   Possible values for the *mode* parameter to the :func:`spawn\* <spawnl>` family of
   functions.  If either of these values is given, the :func:`spawn\*` functions
   will return as soon as the new process has been created, with the process id as
   the return value.

   .. availability:: Unix, Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4148)

Unknown directive type "data".

```
.. data:: P_WAIT

   Possible value for the *mode* parameter to the :func:`spawn\* <spawnl>` family of
   functions.  If this is given as *mode*, the :func:`spawn\*` functions will not
   return until the new process has run to completion and will return the exit code
   of the process the run is successful, or ``-signal`` if a signal kills the
   process.

   .. availability:: Unix, Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4159)

Unknown directive type "data".

```
.. data:: P_DETACH
          P_OVERLAY
```

```
   Possible values for the *mode* parameter to the :func:`spawn\* <spawnl>` family of
   functions.  These are less portable than those listed above. :const:`P_DETACH`
   is similar to :const:`P_NOWAIT`, but the new process is detached from the
   console of the calling process. If :const:`P_OVERLAY` is used, the current
   process will be replaced; the :func:`spawn\* <spawnl>` function will not return.

   .. availability:: Windows.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4171)

Unknown directive type "function".

```
.. function:: startfile(path, [operation], [arguments], [cwd], [show_cmd])

   Start a file with its associated application.

   When *operation* is not specified or ``'open'``, this acts like double-clicking
   the file in Windows Explorer, or giving the file name as an argument to the
   :program:`start` command from the interactive command shell: the file is opened
   with whatever application (if any) its extension is associated.

   When another *operation* is given, it must be a "command verb" that specifies
   what should be done with the file. Common verbs documented by Microsoft are
   ``'print'`` and  ``'edit'`` (to be used on files) as well as ``'explore'`` and
   ``'find'`` (to be used on directories).

   When launching an application, specify *arguments* to be passed as a single
   string. This argument may have no effect when using this function to launch a
   document.

   The default working directory is inherited, but may be overridden by the *cwd*
   argument. This should be an absolute path. A relative *path* will be resolved
   against this argument.

   Use *show_cmd* to override the default window style. Whether this has any
   effect will depend on the application being launched. Values are integers as
   supported by the Win32 :c:func:`ShellExecute` function.

   :func:`startfile` returns as soon as the associated application is launched.
   There is no option to wait for the application to close, and no way to retrieve
   the application's exit status.  The *path* parameter is relative to the current
   directory or *cwd*.  If you want to use an absolute path, make sure the first
   character is not a slash (``'/'``)  Use :mod:`pathlib` or the
   :func:`os.path.normpath` function to ensure that paths are properly encoded for
   Win32.

   To reduce interpreter startup overhead, the Win32 :c:func:`ShellExecute`
   function is not resolved until this function is first called.  If the function
   cannot be resolved, :exc:`NotImplementedError` will be raised.

   .. audit-event:: os.startfile path,operation os.startfile

   .. audit-event:: os.startfile/2 path,operation,arguments,cwd,show_cmd os.startfile

   .. availability:: Windows.

   .. versionchanged:: 3.10
      Added the *arguments*, *cwd* and *show_cmd* arguments, and the
      ``os.startfile/2`` audit event.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4220)

Unknown directive type "function".

```
.. function:: system(command)

   Execute the command (a string) in a subshell.  This is implemented by calling
   the Standard C function :c:func:`system`, and has the same limitations.
   Changes to :data:`sys.stdin`, etc. are not reflected in the environment of
   the executed command. If *command* generates any output, it will be sent to
   the interpreter standard output stream. The C standard does not
   specify the meaning of the return value of the C function, so the return
   value of the Python function is system-dependent.

   On Unix, the return value is the exit status of the process encoded in the
   format specified for :func:`wait`.

   On Windows, the return value is that returned by the system shell after
   running *command*.  The shell is given by the Windows environment variable
   :envvar:`COMSPEC`: it is usually :program:`cmd.exe`, which returns the exit
```

```
      status of the command run; on systems using a non-native shell, consult your
      shell documentation.

      The :mod:`subprocess` module provides more powerful facilities for spawning
      new processes and retrieving their results; using that module is preferable
      to using this function.  See the :ref:`subprocess-replacements` section in
      the :mod:`subprocess` documentation for some helpful recipes.

      On Unix, :func:`waitstatus_to_exitcode` can be used to convert the result
      (exit status) into an exit code. On Windows, the result is directly the exit
      code.

      .. audit-event:: os.system command os.system

      .. availability:: Unix, Windows.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4253)

Unknown directive type "function".

```
   .. function:: times()

      Returns the current global process times.
      The return value is an object with five attributes:

      * :attr:`!user` - user time
      * :attr:`!system` - system time
      * :attr:`!children_user` - user time of all child processes
      * :attr:`!children_system` - system time of all child processes
      * :attr:`!elapsed` - elapsed real time since a fixed point in the past

      For backwards compatibility, this object also behaves like a five-tuple
      containing :attr:`!user`, :attr:`!system`, :attr:`!children_user`,
      :attr:`!children_system`, and :attr:`!elapsed` in that order.

      See the Unix manual page
      :manpage:`times(2)` and :manpage:`times(3)` manual page on Unix or `the GetProcessTimes MSDN
      <https://docs.microsoft.com/windows/win32/api/processthreadsapi/nf-processthreadsapi-getprocesstime
      on Windows. On Windows, only :attr:`!user` and :attr:`!system` are known; the other attributes are

      .. availability:: Unix, Windows.

      .. versionchanged:: 3.3
         Return type changed from a tuple to a tuple-like object
         with named attributes.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4280)

Unknown directive type "function".

```
   .. function:: wait()

      Wait for completion of a child process, and return a tuple containing its pid
      and exit status indication: a 16-bit number, whose low byte is the signal number
      that killed the process, and whose high byte is the exit status (if the signal
      number is zero); the high bit of the low byte is set if a core file was
      produced.

      :func:`waitstatus_to_exitcode` can be used to convert the exit status into an
      exit code.

      .. availability:: Unix.

      .. seealso::

         :func:`waitpid` can be used to wait for the completion of a specific
         child process and has more options.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4298)

Unknown directive type "function".

```
   .. function:: waitid(idtype, id, options)

      Wait for the completion of one or more child processes.
      *idtype* can be :data:`P_PID`, :data:`P_PGID`, :data:`P_ALL`, or
      :data:`P_PIDFD` on Linux.
      *id* specifies the pid to wait on.
      *options* is constructed from the ORing of one or more of :data:`WEXITED`,
      :data:`WSTOPPED` or :data:`WCONTINUED` and additionally may be ORed with
      :data:`WNOHANG` or :data:`WNOWAIT`. The return value is an object
      representing the data contained in the :c:type:`siginfo_t` structure, namely:
      :attr:`si_pid`, :attr:`si_uid`, :attr:`si_signo`, :attr:`si_status`,
      :attr:`si_code` or ``None`` if :data:`WNOHANG` is specified and there are no
      children in a waitable state.

      .. availability:: Unix.

      .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4316)

Unknown directive type "data".

```
.. data:: P_PID
          P_PGID
          P_ALL

   These are the possible values for *idtype* in :func:`waitid`. They affect
   how *id* is interpreted.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4327)

Unknown directive type "data".

```
.. data:: P_PIDFD

   This is a Linux-specific *idtype* that indicates that *id* is a file
   descriptor that refers to a process.

   .. availability:: Linux 5.4+

   .. versionadded:: 3.9
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4336)

Unknown directive type "data".

```
.. data:: WEXITED
          WSTOPPED
          WNOWAIT

   Flags that can be used in *options* in :func:`waitid` that specify what
   child signal to wait for.

   .. availability:: Unix.

   .. versionadded:: 3.3
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4348)

Unknown directive type "data".

```
.. data:: CLD_EXITED
          CLD_KILLED
          CLD_DUMPED
          CLD_TRAPPED
          CLD_STOPPED
          CLD_CONTINUED

   These are the possible values for :attr:`si_code` in the result returned by
   :func:`waitid`.

   .. availability:: Unix.

   .. versionadded:: 3.3

   .. versionchanged:: 3.9
      Added :data:`CLD_KILLED` and :data:`CLD_STOPPED` values.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4366)

Unknown directive type "function".

```
.. function:: waitpid(pid, options)

   The details of this function differ on Unix and Windows.

   On Unix: Wait for completion of a child process given by process id *pid*, and
   return a tuple containing its process id and exit status indication (encoded as
   for :func:`wait`).  The semantics of the call are affected by the value of the
   integer *options*, which should be ``0`` for normal operation.

   If *pid* is greater than ``0``, :func:`waitpid` requests status information for
   that specific process.  If *pid* is ``0``, the request is for the status of any
   child in the process group of the current process.  If *pid* is ``-1``, the
   request pertains to any child of the current process.  If *pid* is less than
   ``-1``, status is requested for any process in the process group ``-pid`` (the
   absolute value of *pid*).

   An :exc:`OSError` is raised with the value of errno when the syscall
   returns -1.

   On Windows: Wait for completion of a process given by process handle *pid*, and
   return a tuple containing *pid*, and its exit status shifted left by 8 bits
   (shifting makes cross-platform use of the function easier). A *pid* less than or
   equal to ``0`` has no special meaning on Windows, and raises an exception. The
   value of integer *options* has no effect. *pid* can refer to any process whose
   id is known, not necessarily a child process. The :func:`spawn\* <spawnl>`
   functions called with :const:`P_NOWAIT` return suitable process handles.

   :func:`waitstatus_to_exitcode` can be used to convert the exit status into an
   exit code.

   .. versionchanged:: 3.5
      If the system call is interrupted and the signal handler does not raise an
      exception, the function now retries the system call instead of raising an
      :exc:`InterruptedError` exception (see :pep:`475` for the rationale).
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4402)

Unknown directive type "function".

```
.. function:: wait3(options)

   Similar to :func:`waitpid`, except no process id argument is given and a
   3-element tuple containing the child's process id, exit status indication,
   and resource usage information is returned.  Refer to
   :mod:`resource`.\ :func:`~resource.getrusage` for details on resource usage
   information.  The option argument is the same as that provided to
   :func:`waitpid` and :func:`wait4`.

   :func:`waitstatus_to_exitcode` can be used to convert the exit status into an
   exitcode.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4417)

Unknown directive type "function".

```
.. function:: wait4(pid, options)

   Similar to :func:`waitpid`, except a 3-element tuple, containing the child's
   process id, exit status indication, and resource usage information is returned.
   Refer to :mod:`resource`.\ :func:`~resource.getrusage` for details on
   resource usage information.  The arguments to :func:`wait4` are the same
   as those provided to :func:`waitpid`.

   :func:`waitstatus_to_exitcode` can be used to convert the exit status into an
   exitcode.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4431)

Unknown directive type "function".

```
.. function:: waitstatus_to_exitcode(status)

   Convert a wait status to an exit code.

   On Unix:

   * If the process exited normally (if ``WIFEXITED(status)`` is true),
     return the process exit status (return ``WEXITSTATUS(status)``):
     result greater than or equal to 0.
   * If the process was terminated by a signal (if ``WIFSIGNALED(status)`` is
     true), return ``-signum`` where *signum* is the number of the signal that
     caused the process to terminate (return ``-WTERMSIG(status)``):
     result less than 0.
   * Otherwise, raise a :exc:`ValueError`.

   On Windows, return *status* shifted right by 8 bits.

   On Unix, if the process is being traced or if :func:`waitpid` was called
   with :data:`WUNTRACED` option, the caller must first check if
   ``WIFSTOPPED(status)`` is true. This function must not be called if
   ``WIFSTOPPED(status)`` is true.

   .. seealso::

      :func:`WIFEXITED`, :func:`WEXITSTATUS`, :func:`WIFSIGNALED`,
      :func:`WTERMSIG`, :func:`WIFSTOPPED`, :func:`WSTOPSIG` functions.

   .. versionadded:: 3.9
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4461)

Unknown directive type "data".

```
.. data:: WNOHANG

   The option for :func:`waitpid` to return immediately if no child process status
   is available immediately. The function returns ``(0, 0)`` in this case.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4469)

Unknown directive type "data".

```
.. data:: WCONTINUED

   This option causes child processes to be reported if they have been continued
   from a job control stop since their status was last reported.

   .. availability:: some Unix systems.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4477)

Unknown directive type "data".

```
.. data:: WUNTRACED

   This option causes child processes to be reported if they have been stopped but
   their current state has not been reported since they were stopped.

   .. availability:: Unix.
```

The following functions take a process status code as returned by :func:`system`, :func:`wait`, or :func:`waitpid` as a parameter. They may be used to determine the disposition of a process.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4485); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4485); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4485); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4489)

Unknown directive type "function".

```
.. function:: WCOREDUMP(status)

   Return ``True`` if a core dump was generated for the process, otherwise
   return ``False``.

   This function should be employed only if :func:`WIFSIGNALED` is true.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4499)

Unknown directive type "function".

```
.. function:: WIFCONTINUED(status)

   Return ``True`` if a stopped child has been resumed by delivery of
   :data:`~signal.SIGCONT` (if the process has been continued from a job
   control stop), otherwise return ``False``.

   See :data:`WCONTINUED` option.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4510)

Unknown directive type "function".

```
.. function:: WIFSTOPPED(status)

   Return ``True`` if the process was stopped by delivery of a signal,
   otherwise return ``False``.

   :func:`WIFSTOPPED` only returns ``True`` if the :func:`waitpid` call was
   done using :data:`WUNTRACED` option or when the process is being traced (see
   :manpage:`ptrace(2)`).

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4521)

Unknown directive type "function".

```
.. function:: WIFSIGNALED(status)

   Return ``True`` if the process was terminated by a signal, otherwise return
   ``False``.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4529)

Unknown directive type "function".

```
.. function:: WIFEXITED(status)

   Return ``True`` if the process exited terminated normally, that is,
   by calling ``exit()`` or ``_exit()``, or by returning from ``main()``;
   otherwise return ``False``.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4538)

Unknown directive type "function".

```
.. function:: WEXITSTATUS(status)

   Return the process exit status.

   This function should be employed only if :func:`WIFEXITED` is true.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4547)

Unknown directive type "function".

```
.. function:: WSTOPSIG(status)

   Return the signal which caused the process to stop.

   This function should be employed only if :func:`WIFSTOPPED` is true.

   .. availability:: Unix.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4556)

Unknown directive type "function".

```
.. function:: WTERMSIG(status)

   Return the number of the signal that caused the process to terminate.

   This function should be employed only if :func:`WIFSIGNALED` is true.

   .. availability:: Unix.
```

Interface to the scheduler

These functions control how a process is allocated CPU time by the operating system. They are only available on some Unix platforms. For more detailed information, consult your Unix manpages.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4572)

Unknown directive type "versionadded".

```
.. versionadded:: 3.3
```

The following scheduling policies are exposed if they are supported by the operating system.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4577)

Unknown directive type "data".

```
.. data:: SCHED_OTHER

   The default scheduling policy.
```

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4581)

Unknown directive type "data".

```
.. data:: SCHED_BATCH

   Scheduling policy for CPU-intensive processes that tries to preserve
   interactivity on the rest of the computer.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4586)

Unknown directive type "data".

```
.. data:: SCHED_IDLE

   Scheduling policy for extremely low priority background tasks.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4590)

Unknown directive type "data".

```
.. data:: SCHED_SPORADIC

   Scheduling policy for sporadic server programs.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4594)

Unknown directive type "data".

```
.. data:: SCHED_FIFO

   A First In First Out scheduling policy.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4598)

Unknown directive type "data".

```
.. data:: SCHED_RR

   A round-robin scheduling policy.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4602)

Unknown directive type "data".

```
.. data:: SCHED_RESET_ON_FORK

   This flag can be OR'ed with any other scheduling policy. When a process with
   this flag set forks, its child's scheduling policy and priority are reset to
   the default.
```

This class represents tunable scheduling parameters used in :func:`sched_setparam`, :func:`sched_setscheduler`, and :func:`sched_getparam`. It is immutable.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4611); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4611); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4611); *backlink*

Unknown interpreted text role "func".

At the moment, there is only one possible parameter:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4617)

Unknown directive type "attribute".

```
.. attribute:: sched_priority

   The scheduling priority for a scheduling policy.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4622)

Unknown directive type "function".

```
.. function:: sched_get_priority_min(policy)

   Get the minimum priority value for *policy*. *policy* is one of the
   scheduling policy constants above.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4628)

Unknown directive type "function".

```
.. function:: sched_get_priority_max(policy)

   Get the maximum priority value for *policy*. *policy* is one of the
   scheduling policy constants above.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4634)

Unknown directive type "function".

```
.. function:: sched_setscheduler(pid, policy, param)

   Set the scheduling policy for the process with PID *pid*. A *pid* of 0 means
   the calling process. *policy* is one of the scheduling policy constants
   above. *param* is a :class:`sched_param` instance.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4641)

Unknown directive type "function".

```
.. function:: sched_getscheduler(pid)

   Return the scheduling policy for the process with PID *pid*. A *pid* of 0
   means the calling process. The result is one of the scheduling policy
   constants above.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4648)

Unknown directive type "function".

```
.. function:: sched_setparam(pid, param)

   Set the scheduling parameters for the process with PID *pid*. A *pid* of 0 means
   the calling process. *param* is a :class:`sched_param` instance.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4654)

Unknown directive type "function".

```
.. function:: sched_getparam(pid)
```

```
   Return the scheduling parameters as a :class:`sched_param` instance for the
   process with PID *pid*. A *pid* of 0 means the calling process.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4660)

Unknown directive type "function".

```
.. function:: sched_rr_get_interval(pid)

   Return the round-robin quantum in seconds for the process with PID *pid*. A
   *pid* of 0 means the calling process.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4666)

Unknown directive type "function".

```
.. function:: sched_yield()

   Voluntarily relinquish the CPU.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4671)

Unknown directive type "function".

```
.. function:: sched_setaffinity(pid, mask)

   Restrict the process with PID *pid* (or the current process if zero) to a
   set of CPUs.  *mask* is an iterable of integers representing the set of
   CPUs to which the process should be restricted.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4678)

Unknown directive type "function".

```
.. function:: sched_getaffinity(pid)

   Return the set of CPUs the process with PID *pid* (or the current process
   if zero) is restricted to.
```

Miscellaneous System Information

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4690)

Unknown directive type "function".

```
.. function:: confstr(name)

   Return string-valued system configuration values. *name* specifies the
   configuration value to retrieve; it may be a string which is the name of a
   defined system value; these names are specified in a number of standards (POSIX,
   Unix 95, Unix 98, and others).  Some platforms define additional names as well.
   The names known to the host operating system are given as the keys of the
   ``confstr_names`` dictionary.  For configuration variables not included in that
   mapping, passing an integer for *name* is also accepted.

   If the configuration value specified by *name* isn't defined, ``None`` is
   returned.

   If *name* is a string and is not known, :exc:`ValueError` is raised.  If a
   specific value for *name* is not supported by the host system, even if it is
   included in ``confstr_names``, an :exc:`OSError` is raised with
   :const:`errno.EINVAL` for the error number.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4711)

Unknown directive type "data".

```
.. data:: confstr_names

   Dictionary mapping names accepted by :func:`confstr` to the integer values
   defined for those names by the host operating system. This can be used to
   determine the set of names known to the system.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4720)

Unknown directive type "function".

```
.. function:: cpu_count()

   Return the number of CPUs in the system. Returns ``None`` if undetermined.

   This number is not equivalent to the number of CPUs the current process can
   use.  The number of usable CPUs can be obtained with
   ``len(os.sched_getaffinity(0))``


   .. versionadded:: 3.4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4732)

Unknown directive type "function".

```
.. function:: getloadavg()

   Return the number of processes in the system run queue averaged over the last
   1, 5, and 15 minutes or raises :exc:`OSError` if the load average was
   unobtainable.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4741)

Unknown directive type "function".

```
.. function:: sysconf(name)

   Return integer-valued system configuration values. If the configuration value
   specified by *name* isn't defined, ``-1`` is returned.  The comments regarding
   the *name* parameter for :func:`confstr` apply here as well; the dictionary that
   provides information on the known names is given by ``sysconf_names``.

   .. availability:: Unix.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4751)

Unknown directive type "data".

```
.. data:: sysconf_names

   Dictionary mapping names accepted by :func:`sysconf` to the integer values
   defined for those names by the host operating system. This can be used to
   determine the set of names known to the system.

   .. availability:: Unix.

   .. versionchanged:: 3.11
      Add ``'SC_MINSIGSTKSZ'`` name.
```

The following data values are used to support path manipulation operations. These are defined for all platforms.

Higher-level operations on pathnames are defined in the :mod:`os.path` module.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4765); *backlink*

Unknown interpreted text role "mod".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4768)

Unknown directive type "index".

```
.. index:: single: . (dot); in pathnames
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4769)

Unknown directive type "data".

```
.. data:: curdir

   The constant string used by the operating system to refer to the current
   directory. This is ``'.'`` for Windows and POSIX. Also available via
   :mod:`os.path`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4776)

Unknown directive type "index".

```
.. index:: single: ..; in pathnames
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4777)

Unknown directive type "data".

```
.. data:: pardir

   The constant string used by the operating system to refer to the parent
   directory. This is ``'..'`` for Windows and POSIX. Also available via
   :mod:`os.path`.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4784)

Unknown directive type "index".

```
.. index:: single: / (slash); in pathnames
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4785)

Unknown directive type "index".

```
.. index:: single: \ (backslash); in pathnames (Windows)
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst`, line 4786)

Unknown directive type "data".

```
.. data:: sep

   The character used by the operating system to separate pathname components.
   This is ``'/'`` for POSIX and ``'\\'`` for Windows.  Note that knowing this
   is not sufficient to be able to parse or concatenate pathnames --- use
   :func:`os.path.split` and :func:`os.path.join` --- but it is occasionally
   useful. Also available via :mod:`os.path`.
```

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-

main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4795)

Unknown directive type "index".

```
.. index:: single: / (slash); in pathnames
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4796)

Unknown directive type "data".

```
.. data:: altsep

   An alternative character used by the operating system to separate pathname
   components, or ``None`` if only one separator character exists.  This is set to
   ``'/'`` on Windows systems where ``sep`` is a backslash. Also available via
   :mod:`os.path`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4804)

Unknown directive type "index".

```
.. index:: single: . (dot); in pathnames
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4805)

Unknown directive type "data".

```
.. data:: extsep

   The character which separates the base filename from the extension; for example,
   the ``'.'`` in :file:`os.py`. Also available via :mod:`os.path`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4811)

Unknown directive type "index".

```
.. index:: single: : (colon); path separator (POSIX)
   single: ; (semicolon)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4813)

Unknown directive type "data".

```
.. data:: pathsep

   The character conventionally used by the operating system to separate search
   path components (as in :envvar:`PATH`), such as ``':'`` for POSIX or ``';'`` for
   Windows. Also available via :mod:`os.path`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4820)

Unknown directive type "data".

```
.. data:: defpath

   The default search path used by :func:`exec\*p\* <execl>` and
   :func:`spawn\*p\* <spawnl>` if the environment doesn't have a ``'PATH'``
   key. Also available via :mod:`os.path`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4827)

Unknown directive type "data".

```
.. data:: linesep
```

```
   The string used to separate (or, rather, terminate) lines on the current
   platform.  This may be a single character, such as ``'\n'`` for POSIX, or
   multiple characters, for example, ``'\r\n'`` for Windows. Do not use
   *os.linesep* as a line terminator when writing files opened in text mode (the
   default); use a single ``'\n'`` instead, on all platforms.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4836)

Unknown directive type "data".

```
.. data:: devnull

   The file path of the null device. For example: ``'/dev/null'`` for
   POSIX, ``'nul'`` for Windows.  Also available via :mod:`os.path`.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4841)

Unknown directive type "data".

```
.. data:: RTLD_LAZY
          RTLD_NOW
          RTLD_GLOBAL
          RTLD_LOCAL
          RTLD_NODELETE
          RTLD_NOLOAD
          RTLD_DEEPBIND

   Flags for use with the :func:`~sys.setdlopenflags` and
   :func:`~sys.getdlopenflags` functions.  See the Unix manual page
   :manpage:`dlopen(3)` for what the different flags mean.

   .. versionadded:: 3.3
```

Random numbers

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4860)

Unknown directive type "function".

```
.. function:: getrandom(size, flags=0)

   Get up to *size* random bytes. The function can return less bytes than
   requested.

   These bytes can be used to seed user-space random number generators or for
   cryptographic purposes.

   ``getrandom()`` relies on entropy gathered from device drivers and other
   sources of environmental noise. Unnecessarily reading large quantities of
   data will have a negative impact on  other users  of the ``/dev/random`` and
   ``/dev/urandom`` devices.

   The flags argument is a bit mask that can contain zero or more of the
   following values ORed together: :py:data:`os.GRND_RANDOM` and
   :py:data:`GRND_NONBLOCK`.

   See also the `Linux getrandom() manual page
   <http://man7.org/linux/man-pages/man2/getrandom.2.html>`_.

   .. availability:: Linux 3.17 and newer.

   .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4884)

Unknown directive type "function".

```
.. function:: urandom(size)

   Return a bytestring of *size* random bytes suitable for cryptographic use.

   This function returns random bytes from an OS-specific randomness source.  The
   returned data should be unpredictable enough for cryptographic applications,
   though its exact quality depends on the OS implementation.

   On Linux, if the ``getrandom()`` syscall is available, it is used in
   blocking mode: block until the system urandom entropy pool is initialized
   (128 bits of entropy are collected by the kernel). See the :pep:`524` for
   the rationale. On Linux, the :func:`getrandom` function can be used to get
   random bytes in non-blocking mode (using the :data:`GRND_NONBLOCK` flag) or
   to poll until the system urandom entropy pool is initialized.

   On a Unix-like system, random bytes are read from the ``/dev/urandom``
   device. If the ``/dev/urandom`` device is not available or not readable, the
   :exc:`NotImplementedError` exception is raised.

   On Windows, it will use ``BCryptGenRandom()``.

   .. seealso::
      The :mod:`secrets` module provides higher level functions. For an
      easy-to-use interface to the random number generator provided by your
      platform, please see :class:`random.SystemRandom`.

   .. versionchanged:: 3.6.0
      On Linux, ``getrandom()`` is now used in blocking mode to increase the
      security.

   .. versionchanged:: 3.5.2
      On Linux, if the ``getrandom()`` syscall blocks (the urandom entropy pool
      is not initialized yet), fall back on reading ``/dev/urandom``.

   .. versionchanged:: 3.5
      On Linux 3.17 and newer, the ``getrandom()`` syscall is now used
      when available.  On OpenBSD 5.6 and newer, the C ``getentropy()``
      function is now used. These functions avoid the usage of an internal file
      descriptor.

   .. versionchanged:: 3.11
      On Windows, ``BCryptGenRandom()`` is used instead of ``CryptGenRandom()``
      which is deprecated.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4928)

Unknown directive type "data".

```
.. data:: GRND_NONBLOCK

   By  default, when reading from ``/dev/random``, :func:`getrandom` blocks if
   no random bytes are available, and when reading from ``/dev/urandom``, it blocks
   if the entropy pool has not yet been initialized.

   If the :py:data:`GRND_NONBLOCK` flag is set, then :func:`getrandom` does not
   block in these cases, but instead immediately raises :exc:`BlockingIOError`.

   .. versionadded:: 3.6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\(cpython-main) (Doc) (library)os.rst, line 4939)

Unknown directive type "data".

```
.. data:: GRND_RANDOM

   If  this  bit  is  set,  then  random bytes are drawn from the
   ``/dev/random`` pool instead of the ``/dev/urandom`` pool.

   .. versionadded:: 3.6
```