

`co_flags.CO_NOFREE` is now always set correctly by the code object constructor based on freevars and cellvars, rather than needing to be set correctly by the caller. This ensures it will be cleared automatically when additional cell references are injected into a modified code object and function.

Yield expressions are now deprecated in comprehensions and generator expressions. They are still permitted in the definition of the outermost iterable, as that is evaluated directly in the enclosing scope.

The repr of deeply nested dict now raises a `RecursionError` instead of crashing due to a stack overflow.

Revert memory allocator changes in the C API: move structures back from `_PyRuntime` to `Objects/obmalloc.c`. The memory allocators are once again initialized statically, and so `PyMem_RawMalloc()` and `Py_DecodeLocale()` can be called before `_PyRuntime_Initialize()`.

Add a new "developer mode": new `"-X dev"` command line option to enable debug checks at runtime.

`SyntaxError` is now correctly raised when a generator expression without parenthesis is used instead of an inheritance list in a class definition. The duplication of the parentheses can be omitted only on calls.

`SyntaxError` is now correctly raised when a generator expression without parenthesis is passed as an argument, but followed by a trailing comma. A generator expression always needs to be directly inside a set of parentheses and cannot have a comma on either side.

A new internal `_Py_SetLocaleFromEnv(category)` helper function has been added in order to improve the consistency of behaviour across different `libc` implementations (e.g. Android doesn't support setting the locale from the environment by default).

Fixed several issues in printing tracebacks (`PyTraceBack_Print()`). Setting `sys.tracebacklimit` to 0 or less now suppresses printing tracebacks. Setting `sys.tracebacklimit` to `None` now causes using the default limit. Setting `sys.tracebacklimit` to an integer larger than `LONG_MAX` now means using the limit `LONG_MAX` rather than the default limit. Fixed integer overflows in the case of more than  $2^{31}$  traceback items on Windows. Fixed output errors handling.

Fix the interactive interpreter looping endlessly when no memory.

Bytearray methods `partition()` and `rpartition()` now accept only bytes-like objects as separator, as documented. In particular they now raise `TypeError` rather of returning a bogus result when an integer is passed as a separator.

BytesWarning no longer emitted when the `fromlist` argument of `__import__()` or the `__all__` attribute of the module contain bytes instances.

Environment variables are once more read correctly at interpreter startup.

Ensure that lexically first syntax error involving a parameter and `global` or `nonlocal` is detected first at a given scope. Patch by Ivan Levkivskyi.

Fixed `OverflowError` in the 'unicode-escape' codec and in `codecs.escape_decode()` when decode an escaped non-ascii byte.

The per-frame tracing logic added in 3.7a1 has been altered so that `frame->f_lineno` is updated before either "line" or "opcode" events are emitted. Previously, opcode events were emitted first, and therefore would occasionally see stale line numbers on the frame. The behavior of this feature has changed slightly as a result: when both `f_trace_lines` and `f_trace_opcodes` are enabled, line events now occur first.

Print the full context/cause chain of exceptions on interpreter exit, even if an exception in the chain is unhashable or compares equal to later ones. Patch by Zane Bitter.

Fix timeout rounding in the select module to round correctly negative timeouts between -1.0 and 0.0. The functions now block waiting for events as expected. Previously, the call was incorrectly non-blocking. Patch by Pablo Galindo.

Prevent crashes when calling methods of an uninitialized `zipimport.zipimporter` object. Patch by Oren Milman.

Standard repr() of `BaseException` with a single argument no longer contains redundant trailing comma.

Fixed a bug in debug memory allocator. There was a write to freed memory after shrinking a memory block.

`PyErr_PrintEx()` clears now the ignored exception that may be raised by `_PySys_SetObjectId()`, for example when no memory.

Two minor fixes for `typing` module: allow shallow copying instances of generic classes, improve interaction of `__init_subclass__` with generics. Original PRs by Ivan Levkivskyi.

PEP 557, Data Classes. Provides a decorator which adds boilerplate methods to classes which use type annotations so specify fields.

The header folding algorithm for the new email policies has been rewritten, which also fixes bpo-30788, bpo-31831, and bpo-32182. In particular, RFC2231 folding is now done correctly.

`io.FileIO.readall()` and `io.FileIO.read()` now release the GIL when getting the file size. Fixed hang of all threads with inaccessible NFS server. Patch by Nir Soffer.

Add `attr:'sys.flags.dev_mode'` flag

Unknown interpreted text role "attr".

The `asyncio.windows_utils.socketpair()` function has been removed: use directly `:func:`socket.socketpair`` which is available on all platforms since Python 3.5 (before, it wasn't available on Windows). `asyncio.windows_utils.socketpair()` was just an alias to `socket.socketpair` on Python 3.5 and newer.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 305); [backlink](#)**

Unknown interpreted text role "func".

warnings: In development (-X dev) and debug mode (pydebug build), use the "default" action for ResourceWarning, rather than the "always" action, in the default warnings filters.

`uuid.getnode()` now preferentially returns universally administered MAC addresses if available, over locally administered MAC addresses. This makes a better guarantee for global uniqueness of UUIDs returned from `uuid.uuid1()`. If only locally administered MAC addresses are available, the first such one found is returned.

Wildcard is now supported in `hostname` when it is one and only character in the left most segment of `hostname` in second argument of `:meth:`ssl.match_hostname``. Patch by Mandeep Singh.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 342); [backlink](#)**

Unknown interpreted text role "meth".

Make `:meth:`msilib.SummaryInformation.GetProperty`` return `None` when the value of property is `VT_EMPTY`. Initial patch by Mark Mc Mahon.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 353); [backlink](#)**

Unknown interpreted text role "meth".

Use `:func:`os.path.expanduser`` to find the `~/.netrc` file in `:class:`netrc.netrc``. If it does not exist, `:exc:`FileNotFoundError`` is raised. Patch by Dimitri Merejkowsky.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 363); [backlink](#)**

Unknown interpreted text role "func".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 363); [backlink](#)**

Unknown interpreted text role "class".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 363); [backlink](#)**

Unknown interpreted text role "exc".

Made `tracemalloc.Traceback` behave more like the `traceback` module, sorting the frames from oldest to most recent. `Traceback.format()` now accepts negative `limit`, truncating the result to the `abs(limit)` oldest frames. To get the old behaviour, one can use the new `most_recent_first` argument to `Traceback.format()`. (Patch by Jesse Bakker.)

Fix wrong usage of `:func:`collections.namedtuple`` in the `:meth:`RobotFileParser.parse()` `<urllib.robotparser.RobotFileParser.parse>` method. Initial patch by Robin Wellner.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 387); [backlink](#)**

Unknown interpreted text role "func".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 387); [backlink](#)**

Unknown interpreted text role "meth".

`:func:'msilib.OpenDatabase'` now raises a better exception message when it couldn't open or create an MSI file. Initial patch by William TisÅcter.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 399); [backlink](#)**

Unknown interpreted text role "func".

`setup()` now warns about invalid types for some fields. The `distutils.dist.Distribution` class now warns when `classifiers`, `keywords` and `platforms` fields are not specified as a list or a string.

Added the `-k` command-line option to `python -m unittest` to run only tests that match the given pattern(s).

Added `nullcontext` no-op context manager to `contextlib`. This provides a simpler and faster alternative to `ExitStack()` when handling optional context managers.

The new `test.support.skip_unless_bind_unix_socket()` decorator is used here to skip `asyncio` tests that fail because the platform lacks a functional `bind()` function for unix domain sockets (as it is the case for non root users on the recent Android versions that run now SELinux in enforcing mode).

`codecs.StreamReader.read(n)` now returns not more than *n* characters/bytes for non-negative *n*. This makes it compatible with `read()` methods of other file-like objects.

The warnings module doesn't leak memory anymore in the hidden warnings registry for the "ignore" action of warnings filters. `warn_explicit()` function doesn't add the warning key to the registry anymore for the "ignore" action.

warnings: When Python is build is debug mode (`Py_DEBUG`), `:exc:'DeprecationWarning'`, `:exc:'PendingDeprecationWarning'` and `:exc:'ImportWarning'` warnings are now displayed by default.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 478); [backlink](#)**

Unknown interpreted text role "exc".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 478); [backlink](#)**

Unknown interpreted text role "exc".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 478); [backlink](#)**

Unknown interpreted text role "exc".

Fixed searching regular expression patterns that could match an empty string. Non-empty string can now be correctly found after matching an empty string.

Added support of splitting on a pattern that could match an empty string.

Fixed issues with binary plists: Fixed saving bytearrays. Identical objects will be saved only once. Equal references will be load as identical objects. Added support for saving and loading recursive data structures.

Drop legacy SSL transport from `asyncio`, `ssl.MemoryBIO` is always used anyway.

`asyncio`: Support `pathlib.Path` in `create_unix_connection`; `sock` arg should be optional

Updates 2to3 to convert from `operator.isCallable(obj)` to `callable(obj)`. Patch by Dong-hee Na.

`inspect.signature` should follow [PEP 8](#), if the parameter has an annotation and a default value. Patch by Dong-hee Na.

Add `time.thread_time()` and `time.thread_time_ns()`

Integers that fit in a signed 32-bit integer will be now pickled with protocol 0 using the INT opcode. This will decrease the size of a pickle, speed up pickling and unpickling, and make these integers be unpickled as `int` instances in Python 2.

Make `asyncio.IncompleteReadError` and `LimitOverrunError` pickleable.

Fixed the looping of `asyncio` in the case of reconnection the socket during waiting `async` read/write from/to the socket.

Restored support of loading marshal files with the `TYPE_INT64` code. These files can be produced in Python 2.7.

Enhance `add_reader/writer` check that socket is not used by some transport. Before, only cases when `add_reader/writer` were called with an `int` FD were supported. Now the check is implemented correctly for all file-like objects.

Fix race condition when flushing a file is slow, which can cause a segfault if closing the file from another thread.

Formally deprecated `aifc.openfp`, `sunau.openfp`, and `wave.openfp`. Since change 7bc817d5ba917528e8bd07ec461c635291e7b06a in 1993, `openfp` in each of the three modules had been pointing to that module's `open` function as a matter of backwards compatibility, though it had been both untested and undocumented.

`cProfile` command line now accepts `-m module_name` as an alternative to script path. Patch by Sanyam Khurana.

Reduce performance overhead of `asyncio` debug mode.

`database` argument of `sqlite3.connect()` now accepts a `:term: 'path-like object'`, instead of just a string.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 665); [backlink](#)**

Unknown interpreted text role "term".

Add Configurable `blocksize` to `HTTPConnection` and `HTTPSConnection` for improved upload throughput. Patch by Nir Soffer.

Add a `cancelled()` method to `:class: 'asyncio.Handle'`. Patch by Marat Sharafutdinov.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 685); [backlink](#)**

Unknown interpreted text role "class".

Fixed determining the MAC address in the `uuid` module: Using `ifconfig` on NetBSD and OpenBSD. Using `arp` on Linux, FreeBSD, NetBSD and OpenBSD. Based on patch by Takayuki Shimizukawa.

Fix potential missed signal in `signal.signal()`.

Fix Blake2 params `leaf_size` and `node_offset` on big endian platforms. Patch by Jack O'Connor.

Add an initializer argument to `{Process,Thread}PoolExecutor`

Fixed compilation of the `socket` module on NetBSD 8. Fixed assertion failure or reading arbitrary data when parse a `AF_BLUETOOTH` address on NetBSD and DragonFly BSD.

Fixed stack corruption in `curses.box()` and `curses.ungetmouse()` when the size of types `chtype` or `mmask_t` is less than the size of `C` long. `curses.box()` now accepts characters as arguments. Based on patch by Steve Fink.

Add 3 new clock identifiers: `:data: 'time.CLOCK_BOOTTIME'`, `:data: 'time.CLOCK_PROF'` and `:data: 'time.CLOCK_UPTIME'`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 757); [backlink](#)**

Unknown interpreted text role "data".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 757); [backlink](#)**

Unknown interpreted text role "data".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 757); [backlink](#)**

Unknown interpreted text role "data".

`plistlib` now catches more errors when read binary plists and raises `InvalidFileException` instead of unexpected exceptions.

Fix the method for checking pad state of `curses WINDOW`. Patch by Masayuki Yamamoto.

Fixed the layout of the `kqueue_event` structure on OpenBSD and NetBSD. Fixed the comparison of the `kqueue_event` objects.

Fixed building the `curses` module on NetBSD.

added required constants to `subprocess` module for setting priority on windows

Remove year (1-9999) limits on the `Calendar.weekday()` function. Patch by Mark Gollahon.

`crypt.mksalt()` now allows to specify the number of rounds for SHA-256 and SHA-512 hashing.

`:func: 'inspect.getfile'` no longer computes the repr of unknown objects to display in an error message, to protect against badly behaved custom reprs.

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

main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 836); [backlink](#)

Unknown interpreted text role "func".

Fix the pthread+semaphore implementation of `PyThread_acquire_lock_timed()` when called with `timeout > 0` and `intr_flag=0`: recompute the timeout if `sem_timedwait()` is interrupted by a signal (EINTR). See also the [PEP 475](#).

Add `mmap.ACCESS_DEFAULT` constant.

Use optimized code for BLAKE2 only with SSSE3+. The pure SSE2 implementation is slower than the pure C reference implementation.

`Calendar.itermonthdates()` will now consistently raise an exception when a date falls outside of the 0001-01-01 through 9999-12-31 range. To support applications that cannot tolerate such exceptions, the new methods `itermonthdays3()` and `itermonthdays4()` are added. The new methods return tuples and are not restricted by the range supported by `datetime.date`.

The `shutil.rmtree()` function has been sped up to 20--40%. This was done using the `os.scandir()` function.

Instances of `pickle.Pickler` subclass with the `persistent_id()` method and `pickle.Unpickler` subclass with the `persistent_load()` method no longer create reference cycles.

Don't release the GIL if we can acquire a multiprocessing semaphore immediately.

Fix `multiprocessing.Process` when `stdout` and/or `stderr` is closed or `None`.

Add `subnet_of` and `superset_of` containment tests to `:class:'ipaddress.IPv6Network'` and `:class:'ipaddress.IPv4Network'`. Patch by Michel Albert and Cheryl Sabella.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 929); [backlink](#)

Unknown interpreted text role "class".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 929); [backlink](#)

Unknown interpreted text role "class".

Remove the `os.stat_float_times()` function. It was introduced in Python 2.3 for backward compatibility with Python 2.2, and was deprecated since Python 3.1.

Add a `subprocess.Popen(text=False)` keyword argument to `subprocess` functions to be more explicit about when the library should attempt to decode outputs into text. Patch by Andrew Clegg.

Add `AbstractEventLoop.sock_recv_into()`.

If nested log adapters are used, the inner `process()` methods are no longer omitted.

The `manager` property on `LoggerAdapter` objects is now properly settable.

Fix timeout rounding in `time.sleep()`, `threading.Lock.acquire()` and `socket.socket.settimeout()` to round correctly negative timeouts between -1.0 and 0.0. The functions now block waiting for events as expected. Previously, the call was incorrectly non-blocking. Patch by Pablo Galindo.

`time.clock()` and `time.get_clock_info('clock')` now emit a `DeprecationWarning` warning.

Extended support for parsing UTC offsets. `strptime('%z')` can now parse the output generated by `datetime.isoformat`, including seconds and microseconds.

traceback: Fix a `TypeError` that occurred during printing of exception tracebacks when either the current exception or an exception in its context/cause chain is unhashable. Patch by Zane Bitter.

Add new function to seal a mock and prevent the automatically creation of child mocks. Patch by Mario Corchero.

Implement the [PEP 564](#), add new 6 new functions with nanosecond resolution to the `:mod:'time'` module:

`:func:'~time.clock_gettime_ns'`, `:func:'~time.clock_settime_ns'`, `:func:'~time.monotonic_ns'`, `:func:'~time.perf_counter_ns'`,  
`:func:'~time.process_time_ns'`, `:func:'~time.time_ns'`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 1043); [backlink](#)

Unknown interpreted text role "mod".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 1043); [backlink](#)



Unknown interpreted text role "func".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 1043); [backlink](#)**

Unknown interpreted text role "func".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 1043); [backlink](#)**

Unknown interpreted text role "func".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 1043); [backlink](#)**

Unknown interpreted text role "func".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 1043); [backlink](#)**

Unknown interpreted text role "func".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 1043); [backlink](#)**

Unknown interpreted text role "func".

2to3 now generates a code that uses abstract collection classes from collections.abc rather than collections.

Prevent a crash when calling the `__init__()` method of a `sqlite3.Cursor` object more than once. Patch by Oren Milman.

Prevent a crash in `sqlite3.Cursor.close()` in case the `Cursor` object is uninitialized. Patch by Oren Milman.

Fix possible crash in `timedelta` constructor called with custom integers.

an empty `asyncio.Queue` now doesn't leak memory when `queue.get` pollers timeout

Allow the flags `re.ASCII`, `re.LOCALE`, and `re.UNICODE` to be used as group flags for regular expressions.

`FutureWarning` is now emitted if a regular expression contains character set constructs that will change semantically in the future (nested sets and set operations).

Added support for the Blowfish hashing in the `crypt` module.

Fix method `set_protocol()` of class `_SSLProtocolTransport` in `asyncio` module. This method was previously modifying a wrong reference to the protocol.

Added a workaround for `getkey()` in curses for ncurses 5.7 and earlier.

Allow use of bytes objects for arguments to `meth:~configparser.ConfigParser.read`. Patch by Vincent Michel.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 1154); [backlink](#)**

Unknown interpreted text role "meth".

Fix `poll.poll([timeout])` in the `select` module for arbitrary negative timeouts on all OSes where it can only be a non-negative integer or -1. Patch by Riccardo Coccioli.

`multiprocessing`'s semaphore tracker should be launched again if crashed.

Make `multiprocessing`'s forkserver process immune to Ctrl-C and other user interruptions. If it crashes, restart it when necessary.

Added support for `AF_UNIX` socket in `asyncio.create_datagram_endpoint`.

Add HTTP/2 status code 421 (Misdirected Request) to `~class:http.HTTPStatus`. Patch by Vitor Pereira.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Misc\NEWS.d\ [cpython-main] [Misc] [NEWS.d] 3.7.0a3.rst, line 1203); [backlink](#)**

Unknown interpreted text role "class".

Added `asyncio.BaseEventLoop.connect_accepted_socket` versionadded marker.

Skip `test_httpservers` `test_undecodable_file` on macOS: fails on APFS.

Skip `test_socket.test_sha256()` on Linux kernel older than 4.5. The test fails with `ENOKEY` on kernel 3.10 (on ppc64le). A fix was merged into the kernel 4.5.

Skip on Android `test_faulthandler` tests that raise `SIGSEGV` and remove the `test.support.requires_android_level` decorator.

The runtime embedding tests have been split out from `Lib/test/test_capi.py` into a new `Lib/test/test_embed.py` file.

`test.support.requires_multiprocessing_queue` is removed. Skip tests with `test.support.import_module('multiprocessing.synchronize')` instead when the semaphore implementation is broken or missing.

Skip `test_get_event_loop_new_process` in `test.test_asyncio.test_events` when `sem_open()` is not functional.

Fix `test_tools.test_unparse`: `DirectoryTestCase` now stores the names sample to always test the same files. It prevents false alarms when hunting reference leaks.

Revert the previous changes, the `if_nameindex` structure is defined by Unified Headers.

Revert the last commit, the `F_LOCK` macro is defined by Android Unified Headers.

Support building Android with Unified Headers. The first NDK release to support Unified Headers is `android-ndk-r14`.

`detect_modules()` in `setup.py` now also searches the sysroot paths when cross-compiling.

Fixes Windows SDK version detection when building for Windows.

Fixes quotes in `PCbuild/clean.bat`

Abort the build when building out of a not clean source tree.

Fixed Argument Clinic sometimes causing compilation errors when there was more than one function and/or method in a `.c` file with the same name.

Update Windows builds to use SQLite 3.21.0.

Update OS X installer to use SQLite 3.21.0.

Record profile-opt build progress with stamp files.

Finish removing support for AtheOS.

Return `None` when `View.Fetch()` returns `ERROR_NO_MORE_ITEMS` instead of raising `MSIError`. Initial patch by Anthony Tuininga.

Fixes Modify button in Apps and Features dialog.

Implement the `Database.Close()` method to help closing MSI database objects.

Make the behavior of `USE_STACKCHECK` deterministic in a multi-threaded environment.

Update macOS installer to use OpenSSL 1.0.2m

Improve tk event exception tracebacks in IDLE. When tk event handling is driven by IDLE's run loop, a confusing and distracting `queue.EMPTY` traceback context is no longer added to tk event exception tracebacks. The traceback is now the same as when event handling is driven by user code. Patch based on a suggestion by Serhiy Storchaka.

Delete unused file `idlelib/tabbedpages.py`. Use of `TabbedPageSet` in `configdialog` was replaced by `ttk.Notebook`.

IDLE: Fix old and new bugs in `pathbrowser`; improve tests. Patch mostly by Cheryl Sabella.

IDLE -- Restrict shell prompt manipulation to the shell. Editor and output windows only see an empty last prompt line. This simplifies the code and fixes a minor bug when newline is inserted. `Sys.ps1`, if present, is read on Shell start-up, but is not set or changed.

The font sample in the IDLE configuration dialog is now editable. Changes persist while IDLE remains open

`Test_code_module` now passes if run after `test_idle`, which sets `ps1`. The code module uses `sys.ps1` if present or sets it to `'>>>'` if not. `Test_code_module` now properly tests both behaviors. Ditto for `ps2`.

Fix a `TypeError` that caused a shell restart when printing a traceback that includes an exception that is unhashable. Patch by Zane Bitter.

Use non-Latin characters in the IDLE's Font settings sample. Even if one selects a font that defines a limited subset of the unicode Basic Multilingual Plane, `tk/tk` will use other fonts that define a character. The expanded example give users of non-Latin characters a better idea of what they might see in IDLE's shell and editors. To make room for the expanded sample, frames on the Font tab are re-arranged. The Font/Tabs help explains a bit about the additions.

Remove CVS and Subversion tools: remove `svneol.py` and `treesync.py` scripts. CPython migrated from CVS to Subversion, to Mercurial, and then to Git. CVS and Subversion are no longer used to develop CPython.

Make redemo work with Python 3.6 and newer versions. Also, remove the `LOCALE` option since it doesn't work with string patterns in Python 3. Patch by Christoph Sarnowski.

Fix `PyGILState_Ensure()`. When `PyGILState_Ensure()` is called in a non-Python thread before `PyEval_InitThreads()`, only call `PyEval_InitThreads()` after calling `PyThreadState_New()` to fix a crash.

The `Py_UseClassExceptionsFlag` flag has been removed. It was deprecated and wasn't used anymore since Python 2.0.

Move the current exception state from the frame object to the co-routine. This simplifies the interpreter and fixes a couple of obscure bugs caused by having swap exception state when entering or exiting a generator.

Add `Py_RETURN_RICHCOMPARE` macro to reduce boilerplate code in rich comparison functions.

The `PyExc_RecursionErrorInst` singleton is removed and `PyErr_NormalizeException()` does not use it anymore. This singleton is persistent and its members being never cleared may cause a segfault during finalization of the interpreter. See also issue #22898.