IDA Pro 8.0 Full list of changes and new features:

Procesor modules:

- 68K: support switches which use cmpa for the range check
- ARM: improve handling of manual setting of ARM/Thumb mode via the T pseudo-register
- AVR: added config for ATmega640
- PC: improve function recognition

Debuggers:

- PIN: support PIN 3.22-98547

File formats:

- COFF: support ARM64 and ARMv7 object files compiled with /bigobj option
- DWARF: upgrade libdwarf to version 20220625 (aka 0.4.1)
- MACHO: improve symbolication of branch mappings in iOS16+ dyldcaches
- MACHO: support for iOS16 dyld caches
- MACHO: when loading a dyld shared cache, make "single module" option the default choice

FLIRT / TILS / IDS:

- FLIRT: GO: increased coverage of golang signatures
- FLIRT: MFC: added signatures for vc1431 (Visual Studio 16.11.10)
- FLIRT: VC: added signatures for vc1431 32bit(Visual Studio 16.11.10)
- idaclang: added "--idaclang-parse-static" option to the cmdline tool
- idaclang: introduced the "--idaclang-extra-c-mangling" option for building type libs for mixed-language inputs (e.g. C++, C, and Objective-C)
- idaclang: try to pre-set a default target configuration that corresponds to the currently loaded file

Standard plugins

- DSCU: support loading (and symbolicating) global offset tables from iOS16
- dyldcaches
- golang: support for go1.18 (function names, types)
- OBJC: improved decompilation of functions that use objc_alloc_init() to initialize Objective-C objects
- OBJC: improved decompilation of Objective-C binaries by creating artificial imports for to methods not present in the idb
- patfind: new plugin to discover code patterns in otherwise unmarked binaries

Scripting & SDK

- IDAPython: removed Python 2 support
- SDK: added a new method qstring::rtrim() to trim whitespaces
- SDK: added get_stdact_descs() for choosers for customizing the standard actions (Insert, Delete, Edit, Refresh)
- SDK: added wildcard_path_match(), that can match entire paths against a pattern following the same rules as a shell (e.g. ** and ranges like [a-z])
- SDK: improved comment for has_external_refs()
- SDK: support usage of qstring in hashed STL containers

UI:

- UI: the command-line arguments in the Debugger>Process options... dialog are no longer limited to 1024 characters

Decompilers:

- added option HO_PROP_VOLATILE_LDX to propagate load instructions without checking for volatile memory access
- added support for outlined functions
- arm: recognize thunk functions with suffixes _from_thumb, _from_arm, _veneer
- improve handling of scattered return values (=using mutiple registers/stack locations)
- new decompiler: HEXARC (for the ARC processor family)
- pc: control register maniplation intrinsics (e.g __writecr0) work with 32-bit values in 32-bit mode
- support WCHAR, wchar16_t, wchar32_t as character element types

Bugfixes:

BUFGIX: IDC: definitions of SN_CHECK/SN_NOCHECK (flags for set_name()) were wrong

BUGFIX: ARM: fixed an endless loop which could occur when analyzing code switching between ARM/Thumb modes

BUGFIX: ARM: IDA could display a "bad instruction decoding" warning when trying to decode an undefined instruction

BUGFIX: ARM: some undefined A64 instructions were wrongly decoded as FCMEQ

BUGFIX: ARM: arm64 function arguments with wrong attributes could crash ida

BUGFIX: automatically created string literal names would have repeating symbols in place of embedded zeroes in the string

BUGFIX: dbg: IDA could produce an internal error when undo was used during debugging

BUGFIX: decompiler: do not crash if nullptr is passed to various save_.. functions

BUGFIX: decompiler: do not optimize away successive volatile memory reads

BUGFIX: decompiler: fix sometimes wrong decompilation when loading values from memory in big-endian mode

BUGFIX: decompiler: fixed multiple interrs

BUGFIX: decompiler: modifies_d() was incorrectly returning true for instructions without the 'd' operand

BUGFIX: DWARF: during source-level debugging, location of some items wouldn't be properly resolved

BUGFIX: DWARF: The plugin could INTERR because of how duplicate types were handled

BUGFIX: golang: IDA could hang when parisng metadata in some Go binaries

BUGFIX: IDA could crash when loading PE files if IDS debugging was enabled (-z40 switch)

BUGFIX: IDA could fail to load bytes from modules in iOS 15 dyldcaches for older iphones (iphone X and earlier)

BUGFIX: IDA could fail to load symbols for some modules in iOS 15 dyldcaches

BUGFIX: idaclang could create invalid types after parsing a "using" declaration that has the same name as an existing type

BUGFIX: idaclang could fail to parse c++ type declarations that use the "auto" keyword

BUGFIX: idaclang would fail to parse function prototypes that have an unspecified number of arguments

BUGFIX: IDAPython: fixed multiple crashes and infinite loops when wrong arguments are passed to IDA APIs

BUGFIX: IDAPython: IDA could crash if 'has_insn_feature' was called with improper data

BUGFIX: IDAPython: internal errors in IDA API wrappers which are called bypassing IDA UI (e.g. from alternative IDAPython shells) are now caught and reported properly

BUGFIX: IDAPython: when trying to create a too big segment, produce a warning instead of fatal error

BUGFIX: IDC: calling get_tev_reg() with wrong data could produce "No error" message instead of showing the correct error

BUGFIX: installer: PIN debugger plugin was not shipped with Mac builds of IDA by mistake

BUGFIX: kernel: compact_numbered_types() was mishandling aliased types

BUGFIX: kernel: fixed an endless loop which could occur during application of startup signatures

BUGFIX: kernel: fixed interr 641 that could occur when parsing a bad function prototype

BUGFIX: kernel: get_strlit_contents() could loop very long time even when maxcps was set to a reasonable value

BUGFIX: kernel: IDA could produce "database corrupted" when undoing some operations

BUGFIX: MACHO: some ARM64e binaries could have wrong pointer values, leading to wrong parsing of Objective-C metadata

BUGFIX: MIPS: bltzal and bgezal were not handled as call instructions

BUGFIX: OBJC: "Run until message received" action could fail on macOS 12

BUGFIX: PC: some 64-bit functions would lose offsets when Lumina metadata was applied

BUGFIX: PC: ud1 instruction was decoded incorrectly (the mod r/m byte was not parsed)

BUGFIX: PDB: fixed interr 984 which could occur when loading PDBs with types from recent Windows builds

BUGFIX: PDB: the PDB file download could be cancelled unexpectedly when using symsrv.dll from WinDbg Preview

BUGFIX: PPC: functions using 'ba' for tail calls to noret functions were not marked as noret

BUGFIX: SDK: get_name_ea() would return non-BADADDR results for structure or enum names

BUGFIX: svdimport: plugin could crash when processing certain SVD files

BUGFIX: tilib: fixed interr 157 that could occur when listing til contents in the presence of type aliases

BUGFIX: UI: database snapshots were added to the recent files list and could fill it completely

BUGFIX: UI: IDA could produce internal errror 40225 after some user manipulations with the function graphs

BUGFIX: UI: IDA would not display shortcuts for actions in context menus on macOS

BUGFIX: UI: strings containing \r\n could be printed as empty in the Output window and the log file

BUGFIX: UI: TOOL_CLOSED_BY_ESC in idagui.cfg did not work

BUGFIX: windbg: IDA could crash if a breakpoint it added became invalid (e.g. by user's actions bypassing IDA's UI)

IDA 8.0.220829(sp1)

Bugfixes

- BUGFIX: debugging was broken on macOS13
- BUGFIX: IDA could crash on some systems when checking for new version from the UI
- BUGFIX: IDA could fail to symbolicate stubs in iOS16 dyldcaches
- BUGFIX: IDA could freeze (on linux) when the "Open file" dialog was opened during autoanalysis
- BUGFIX: IDA could load some bytes incorrectly from iOS16 dyldcaches

- BUGFIX: IDA could show a "Running" dialog when stepping over/into during debugging with GDB/iOS/WinDbg
- BUGFIX: IDA would crash when using "Create struct from data" if struct_0 already existed
- BUGFIX: IDA would fail to generate nice names for __objc_stubs sections in stripped macOS12 binaries
- BUGFIX: iOS debugger was broken on iOS 16
- BUGFIX: kernel: IDA would incorrectly move function bounds if it started with a nop instruction
- BUGFIX: PIC: IDA would crash when trying to change the current device for the PIC18 family via the global Options dialog
- BUGFIX: RX: some variants of the movu instruction were decoded incorrectly (wrong displacement value)
- BUGFIX: ui: in the Load new file dialog, processors and processor families were sorted case-sensitively, making it harder to discover some of them
- BUGFIX: decompiler: decompiler could hang when decompiling some functions with merged calls optimization
- BUGFIX: decompiler: fix interr 50458 which could occur in some MIPS binaries