

The contents of this file are subject to the GNU General Public License (GPL) Version 2 or later (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.gnu.org/copyleft/gpl.html>

Software distributed under the License is distributed on an "AS IS" basis, without warranty of any kind, either expressed or implied. See the License for the specific language governing rights and limitations under the License.

This file was originally developed as part of the software suite that supports the book "The Elements of Computing Systems" by Nisan and Schocken, MIT Press 2005. If you modify the contents of this file, please document and mark your changes clearly, for the benefit of others.

2.5

===

General

Added: Completely redesigned web site. The site has a new look & feel and contains more information in less text.

Added: Lectures for all projects to the web site.

Compiler

Added: Error when returning a value from a void function or returning with no value from a non-void function.

Added: Error when the program flow could reach past the last statement of a subroutine (without being obstructed by a 'return' statement).

Added: Warning when unreachable statements are encountered.

Added: All compiler messages now have a uniform look which also includes the current class and subroutine name (in addition to the description and line number).

Added: Most of the compiler code has been reorganized so that upon an error parsing always continues (upon all semantic errors parsing continues straight away; upon most syntax errors a reasonable correction is guessed and parsing continues straight away as well; parsing continues from a near point otherwise) - this way the vast majority of errors are reported in one program run.

Added: Upon a compilation failure on a file, the compilation process does not stop - compilation of the next file is attempted. Subroutine calls are only cross-referenced with their signatures if the file containing the subroutine definition was successfully compiled (otherwise no cross-referencing is done, as if the erroneous file was not given as an argument to the compiler). This is done to avoid massive errors upon problems with the subroutine (and not with calls).

Added: Upon a compilation error, the generated file is deleted. The file is still generated upon a "link"-style error (error when cross-referencing subroutine signatures and calls).

Added: Clearer, far more indicative error messages on many common Jack programming mistakes.

Fixed: Line numbers in the source code reported in errors/warnings are now always correct (the line number of the next token was previously reported in most cases, and in the rest of the cases yet even a higher number than that of the next token was reported).

Added: Error if a source file contains any code beyond the closing } of the class definition (previously this was silently ignored)

Added: When run with no parameters, a about/usage message is printed. This message is consistent with the about/usage dialog boxes of the graphic applications of the software suite.

Fixed: An error when declaring two subroutines (of any kinds) of the same name in the same class is now always issued when needed.

2.0

===

General

Fixed: .bat/.sh files used to launch the programs now work also when activated from a prompt from a different directory. These files also now work correctly when there's a file name with spaces in the classpath (like when the JRE on windows was installed into "Program Files").

Fixed: In all simulator/emulators, the while command now behaves as it should. Previously the body of the while was ignored which resulted in a busy wait for nothing. So far this command has only been used in Memory.tst and this was overlooked there because the memory polling worked after a while even though the "eval" command was skipped but this created a very heavy CPU load.

Fixed: The "official" version number is now the same as the one displayed by all programs.

Added: All zip archives are now compressed with the "best" compression, making all of them considerably smaller.

Added: Added FAQ and errata sections to the web site.

Projects

Fixed: Numerous file name errors in scripts (case errors, suffix errors), most of which caused tests not to compile or not to work on unix platforms.

Fixed: Some project files on the site were different in their zipped version than in their non-zipped version. All such discrepancies have been synchronized and are being kept synchronized with a set of scripts.

Added: For better ease-of-use, project 1 tst and cmp files do not contain any hexadecimal values anymore.

Added: Project 3 Bit test now catches many more bad implementations.

Added: Project 3 Register/RAM* tests have been systematically revamped to catch many more bad implementations - most are now around a 1000 lines in length.

Fixed: Project 5 CPU.cmp would fail on a good chip that was implemented differently. Fixed by using wildcards for outM when writeM is false.

Added: Alternative, not recommended, ess thorough implementations for some project 5 tests which do not require the use of specific built-in chips.

Fixed: Bat destructor was not called in Project 11 PongGame.jack.

Added: Project 12 StringTest now also tests a zero-capacity string.

Fixed: Project 12 StringTest now tests setInt regardless of whether the implementation of Output.printInt calls it. Also, just to be on the safe side, appendChar is tested also in the remote case of a student-written compiler which doesn't use appendChar for string literals (highly unlikely - how would the length of a string be incremented?).

Assembler -----

Fixed: Spaces are now correctly ignored in the comp part of a C-instruction.

Fixed: An error is reported within the program if the loaded comparison file contains non-0/1 characters (previously an exception was thrown).

Fixed: An error is reported within the program if a line in the loaded comparison file is not of length 16 (previously the file was loaded and the lines were compared numerically so the comparison would even succeed in certain cases).

Added: The version number is now displayed in a fashion consistent with the rest of the software suite.

Hardware Simulator: -----

Added: Detailed documentation for the Chip API in the Readme.txt file of the open source distribution.

Added: Support for wildcards in compare files of the Hardware Simulator. This was added to allow fixing of the CPU.cmp test file.

Fixed: No check was performed that a gate definition was closed (}). This means that no errors were issued when a /* comment was not closed or

after loading a partial or incomplete HDL file.

Fixed: The binary input component now appears in the correct location with the correct number of active bits. Also, double-clicking on another input pin while the binary input component resulted in the wrong input being updated - this is now fixed with a new behaviour which has the same feel as the decimal input: clicking on another input pin now closes a visible binary input component while double clicking on another input pin now opens the binary input component for that pin. Also enter and escape presses now correspond to pressing the ok/cancel buttons to complete the correct feel for this component.

Added: Clearer error when a script attempts to directly access a built-in chip that's not loaded (like the when using the project 5 scripts accessing PC[] or ARegister[] when these have not been used).

CPU Emulator:

Fixed: Negative values may now be stored in the A register as long as no attempt to dereference them or to jump to them is made.

VM Emulator:

Added: Support for built-in implementation of VM functions (similar to built-in gates). The API enabling this (the VMCode API) is meticulously documented in the Readme.txt file of the open source distribution. Via this API, a built-in implementation for all Jack OS classes was added - they can work in any combination of built-in/*.vm classes. This yields a noticeable speedup in running Jack-OS-based VM applications, especially in graphics drawing (no need to make the space invaders look like squares anymore :)) - it is advisable to add Sys.wait calls to code that runs too fast now (try running Pong now - it's like running the XT version of Digger on a Pentium). How it works: Load a directory which doesn't contain a *.vm implementation for the Jack OS classes. A dialog will pop up to confirm the usage of built-in implementations - click 'Yes'. Now run everything normally - when encountering a call to a function in a class with no *.vm implementation and if that function has a built-in implementation, it will be invoked. If data flow animation is on, the changing of heap/screen memory by built-in functions will be animated. The built-in implementation of Sys.error and Sys.halt now offer the side-effect of a pop-up window with information regarding the reason for the program halt.

Fixed: Incorrect size of local segment and of stack when returning from a function called when more values than its arguments were on the stack.

Fixed: Incorrect detection (both false positive and false negatives) of encountering a 'function' statement in the middle of a function definition.

Fixed: Memory leak when restarting a program from inside a function (stackFrames vector not cleared).

Fixed: Error regarding static segment if Sys.init accidentally contains a

return statement (and not enters an endless loop).

Added: Added an error when loading a program consisting of multiple files when there is no implementation of Sys.init.

Added: Now support c-style /* */ comments (because the programs on pages 138 and 140 accidentally contained such comments).

Operating System:

Fixed: The cursor is now displayed when expecting input, as described in the book.

Fixed: Backspace now wraps from the beginning of a line to the previous, providing a more natural behaviour when the user inputs a string to the Keyboard.readLine function.

Compiler:

Added: Error when calling a method of the current class from within a function implicitly, as if for the "this" object.

Added: Error when calling a function or constructor of a known (passed to the compiler for compilation, regardless of the compilation order) class as a method (both from the same class for the "this" object implicitly, and for non-this objects of a known class explicitly).

Added: Error when calling a method of a known class as a function/constructor.

Added: Error when calling a non-existent function/method/constructor of a known class.

Added: Error when declaring two subroutines (of any kinds) of the same name in the same class.

Added: Error when calling a subroutine of a known class with a different number of arguments than the declared number of arguments.

1.0

===

First public release - released with the book.