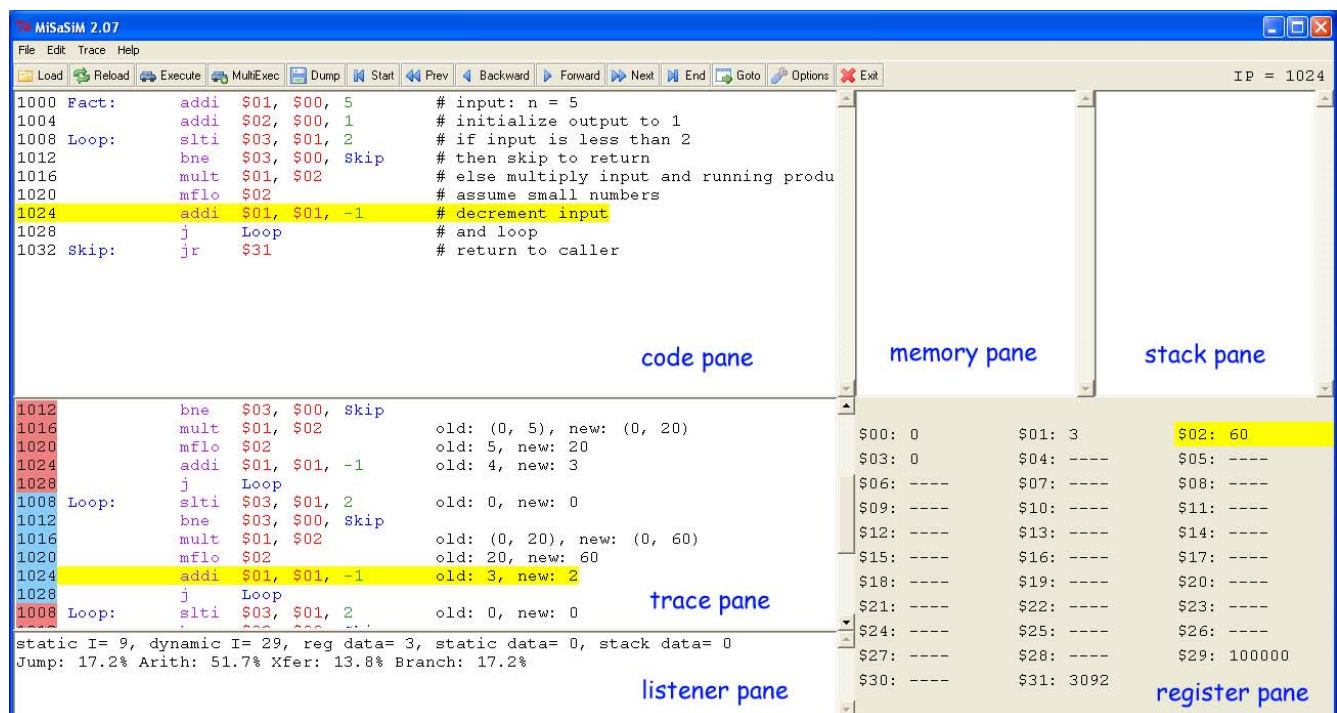




## A Trace Level Assembly Simulator

MiSaSiM is a simulator for exploration of the functionality and efficiency of MIPS assembly programs. It employs a trace to allow both forward and backward navigation of a program's execution. It also provides operating system calls (through software interrupts) for data initialization, visualizations, etc. It runs under python for maximum platform independence, compact code size, and a rich user interface. This document outlines MiSaSiM's key features and their use. For installation instructions, go to [www.ece.gatech.edu/~scotty/misasim/](http://www.ece.gatech.edu/~scotty/misasim/).

When MiSaSiM is launched, the main window opens. In this example, the factorial example "fact.asm" is loaded.



### Window Panes

The main window is segmented into the following panes.

**Code pane:** displays the source code that is being simulated, with the instruction address shown to the left of each instruction.

**Trace pane:** displays the series of instructions that have been executed and the changes in the value of each instruction's destination register (if it has one). As the

user navigates through the trace, the yellow highlighted line shows the current instruction in the trace pane and the corresponding instruction in the code pane. The instruction addresses are also displayed to the left of the instructions, with color coding to indicate changes in control flow (e.g., at loop body or subroutine boundaries or when branches are taken).

**Memory pane:** shows the current state (Address: Value) of data words in static or heap memory that have been written so far.

**Stack pane:** shows the current state (Address: Value) of data words in stack memory. When the stack is popped, the popped values are grayed out.

**Register pane:** shows the current values in the register file, with the most recently changed value highlighted in yellow.

**Listener pane:** shows statistics on the executed code, including static instruction count, dynamic instruction count, number of registers used, memory usage, and dynamic instruction mix (i.e., what percentages of different types of instructions occurred in the trace). This pane also shows error and warning messages.

## Command Buttons

The command buttons have the following functions.

**Load:** load a MIPS assembly file.

**Reload:** reload the currently loaded file. Since the source file is typically edited using notepad, emacs, etc., this command loads the current file version.

**Execute:** execute the loaded assembly file generating a new trace for instructions executed. This command is automatically executed if the navigation buttons are selected before execution has taken place.

**MultiExec:** This command executes the loaded program multiple times. The number of executions is user specified. This is useful for computing an average execution time.

**Dump:** This command dumps the current memory map into a user selected text file. This file contains both memory address and data pairs. Note that the memory values recorded are for the selected trace position.

**Start:** This command positions the trace at the initial executed instruction (i.e., the trace start).

**Prev:** This command positions the trace at the last encountered occurrence of the current instruction. This is useful for stepping back to a previous iteration of a loop.

**Backward:** This command moves backwards in the trace to the instruction executed immediately prior to the current instruction.

**Forward:** This command moves forwards in the trace to the instruction executed

immediately after to the current instruction.

**Next:** This command positions the trace at the next encountered occurrence of the current instruction. This is useful for stepping forward to the next iteration of a loop.

**End:** This command positions the trace at the final executed instruction (i.e., the trace end).

**Goto:** This command positions the trace at the next occurrence of the user specified address.

**Options:** This command allows the adjustment of the maximum instruction limit. This value prevents simulator lockup when infinite loops are encountered. Other options include setting font size and face, and button labeling.

**Exit:** this command exits MiSaSiM.