

Base Class UVSim - Does the overall managing of the program, coordinates between the memory and the accumulator

Attributes:

memory: Memory

accumulator: Accumulator

Methods:

run(): Calls the function to run the instructions of the program that have been loaded in. No input parameters or return values. Results in the program being run.

__execute_instruction(): Executes the instructions for the program. No input parameters or returned values. This function has all the instruction numbers to call the various command functions.

read(): Read a word from the keyboard into a specific location in memory. Return inputted number. Returns the number that is taken in from the keyboard and returns it. Post function is a returned number to be used in a specific location in memory.

reset(): Resets all of the values inside of the program so that the user can start fresh with new values.

initialize(): Sets all of the numeric variables to zero.

get_log(): Returns the log of commands back to the gui.

load_file(): Opens the txt file that the user decides to input. From there it goes through the file and vets all the numbers to make sure that the file is valid and will work. If everything is working correctly then the values are loaded into the registers that were created.

write(registers, target): Write a word from a specific location in memory to screen. Input parameters are the target register that the user wants to put on the screen as well as all the registers to sort through. Target number is then printed to the screen for the user to see. Function results in the ability to see a specific number that is chosen.

store(accumulator): Store a word from the accumulator into a specific location in memory. The input parameter is the accumulator in which will be stored into a piece of memory that is in the instruction. The return for this function will be the number to store into memory. Results of this function are a piece of memory that is assigned a number, or has its number changed.

load(registers, target): Load a word from a specific location in memory into the accumulator. The input parameters are the registers to sort through and then the target register to pull from. The returned value is a number that is then taken and stored into the accumulator. This function results in the accumulator being changed.

Class App - Holds the basic structure for the GUI.

Methods:

setup_frames(): Sets up the main layout frames for the gui. This function takes no parameters and returns nothing but gives a good base for the structure.

setup_left_frame(): Configures the left frame grid. This frame holds the registers as well as the buttons to Step, Run, and Pause the program. No parameters and nothing returned but sets up the main portion of gui.

setup_right_frame(): Configures the right frame grid. This frame holds the file information as well as the value information such as the accumulator and the current register. No parameters and nothing is returned.

create_menu(): Creates a menu for the user to be able to select certain actions that they want to take.

run(): Saves the edits from the user and then runs the program by going through each of the commands

save_edits(): Saves the changes that the user put directly into the gui window.

save_file(): Saves the file, makes sure that the new saved file does not go over the 250 register limit.

file_input(): Loads a file. Takes as input a file name. Reads the file and returns the commands loaded into the dictionary. Registers is an empty dictionary before the function runs and is filled after the function is complete.

reset(): Resets the file, clears all commands and data. Does not take anything in and does not return anything. Results in a clear table and a fresh start for the program.

update_values_display(): Updates the display that shows the accumulator and the current register. No parameters to take in and no returned value. Simply just updates the numbers in the gui after being called.

rotate_color(): Gets the new color from the user and then changes the color of the entire gui.

get_text_color(): Gets the inputted new color from the user.

open_settings(): Opens the settings box to change the color.

reset_colors(): Resets the colors back to the UVU original color scheme.

valid_hex(): makes sure the hex value that the user inputted is valid and works.

load_color_scheme(): Loads the new colors into the json file.

update_values_display: Updates the text box with the accumulator and current register values.

update_console_log(): Updates the console that stores all of the commands that have been logged.

update_memory_display(): Updates the registers memory when they are needing to be updated.

set_accumulator_value(): Updates the value for the accumulator on the gui side. Takes in the accumulator in the UVSim file as a parameter and then updates the number value. Does not update the gui.

set_register_value(): Updates the value for the current register on the gui side. Takes in the current register in the UVSim file as a parameter and then updates the number value. Does not update the gui.

write_to_accumulator(): Writes the content of the accumulator in the gui to the accumulator variable. Uses the content of the gui accumulator as a parameter input and then sets the value. No return value given. Post function results in an updated value by the user.

read_from_gui(): Takes in the input from the gui. Uses functionality from Tkinter to find out what was inputted in the text box. No return value or parameters. Post function results in a variable that can be used in many different areas for different values.

new_window(self): Creates a new object of the class when the “new window” button is selected. Creates a new window that can operate on its own.