Project 2 milestone Design Document

Overview:

The UVSim is a program designed to simulate a virtual machine. The main purpose is to educate computer science students on machine language and computer architecture. The simulator will interpret lines of machine language in BasicML. Instructions are read from a file into the program.

User Stories:

As a computer science student

I want to learn computer architecture

and get an understanding of machine language

As a computer science teacher

I want to educate my students

on machine language

Use Cases:

1: Executing BasicML Code

Actor: Student

System: UVSim

Goal: Execute machine language instructions from a file.

Steps:

The student enters BasicML code into a text file.

The student loads the file into UVSim.

The system reads the instructions from the file.

The system executes the instructions.

The program runs until completion.

2: Loading Data from Memory

Actor: User

System: UVSim

Goal: Load data from a specific memory location into the accumulator.

Steps:

The user inputs the LOAD instruction with a memory address (e.g., 2020).

The system retrieves the value stored in the specified memory location.

The system loads the value into the accumulator.

3: Performing Addition

Actor: John

System: UVSim

Goal: Add a number to one stored in memory.

Steps:

John inputs a number into memory using READ 1043.

John loads the number into the accumulator using 2043.

John adds the number stored in memory location 21 using 3021.

The system updates the accumulator with the sum.

4: Writing Input to Screen

Actor: Teacher

System: UVSim

Goal: Demonstrate writing words from memory to the screen.

Steps:

The teacher inputs a word using READ 1002.

The system stores the word in memory.

The teacher uses WRITE 1102 to display the word.

The system outputs the word to the screen.

5: Performing Division

Actor: Emily

System: UVSim

Goal: Divide a number by a stored value.

Steps:

Emily inputs a number into memory using 1009.

Emily loads the number into the accumulator using 2009.

Emily divides by the number stored at memory location 08 using 3208.

The system updates the accumulator with the result.

6: Reading Input into Memory

Actor: Student

System: UVSim

Goal: Store keyboard input into memory.

Steps:

The student inputs 1001 to read a value from the keyboard.

The system stores the value in memory.

7: Halting the Program

Actor: Tester

System: UVSim

Goal: Ensure the program stops execution properly.

Steps:

The tester inputs 4300 at the end of the program.

The system halts execution and exits.

8: Branching to a Different Memory Location

Actor: Programmer

System: UVSim

Goal: Redirect execution to a different memory location.

Steps:

The programmer inputs 4026.

The system jumps to memory location 26.

Execution continues from that location.

9: Storing a Value in Memory

Actor: David

System: UVSim

Goal: Store a value into memory.

Steps:

David enters 1001 to store a word into memory location 01.

The system stores the word in memory.

10: Performing Subtraction

Actor: Sarah

System: UVSim

Goal: Subtract two numbers.

Steps:

Sarah inputs 1001 to store the first number in memory location 01.

Sarah inputs 1002 to store the second number in memory location 02.

Sarah loads the second number into the accumulator using 2002.

Sarah subtracts the first number using 3101.

The system updates the accumulator with the result.