Group members: Ikhlas Ahmed Khan

Musab

Ramail

Bilal Adnan

Running the following codes;

**INPUTS**;

**vetvli ra, zero, e32,m1,tu,mu**

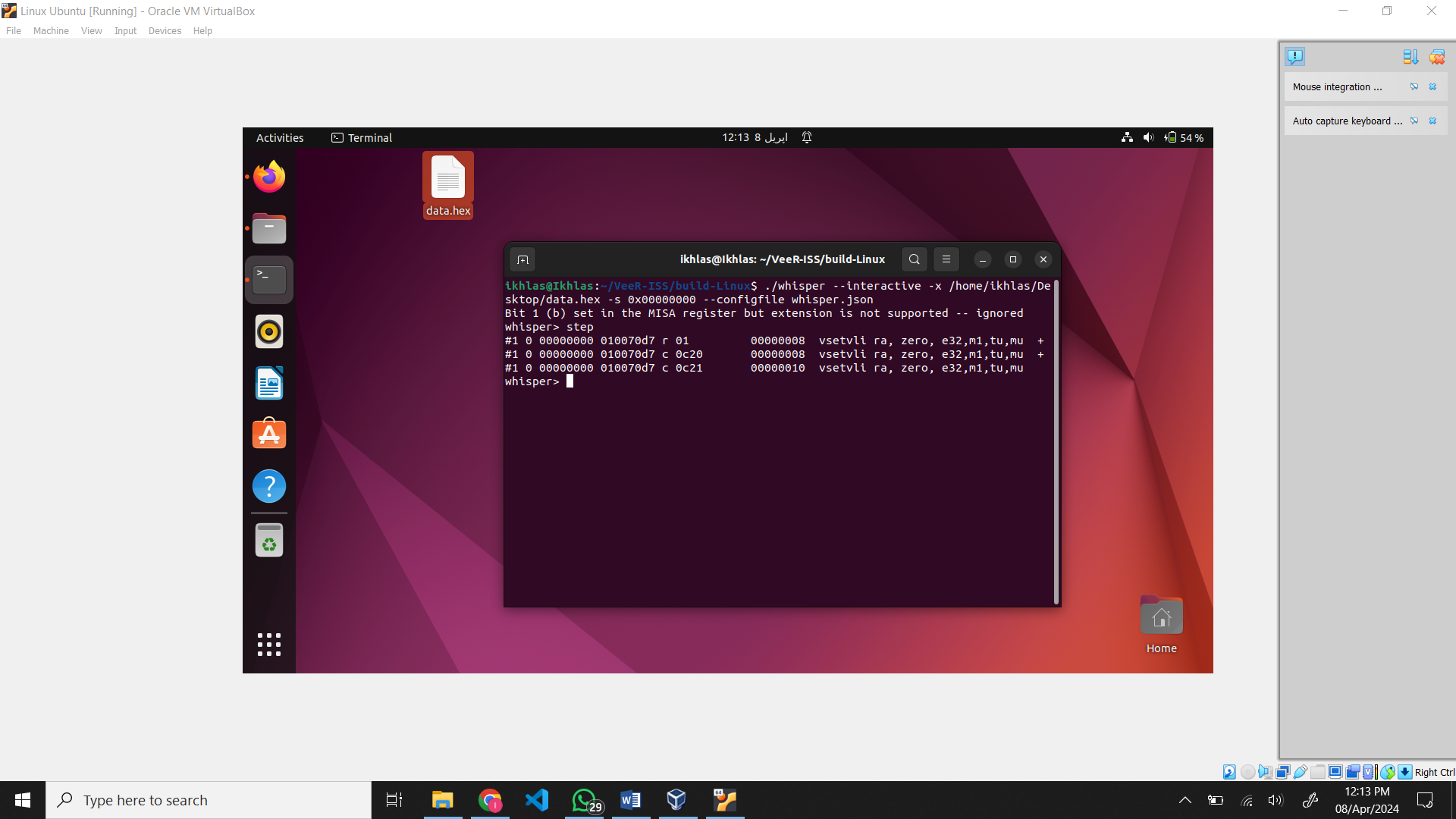
**vadd.vi v1, v0, 0x5**

**vadd.vi v2, v1, 0x6**

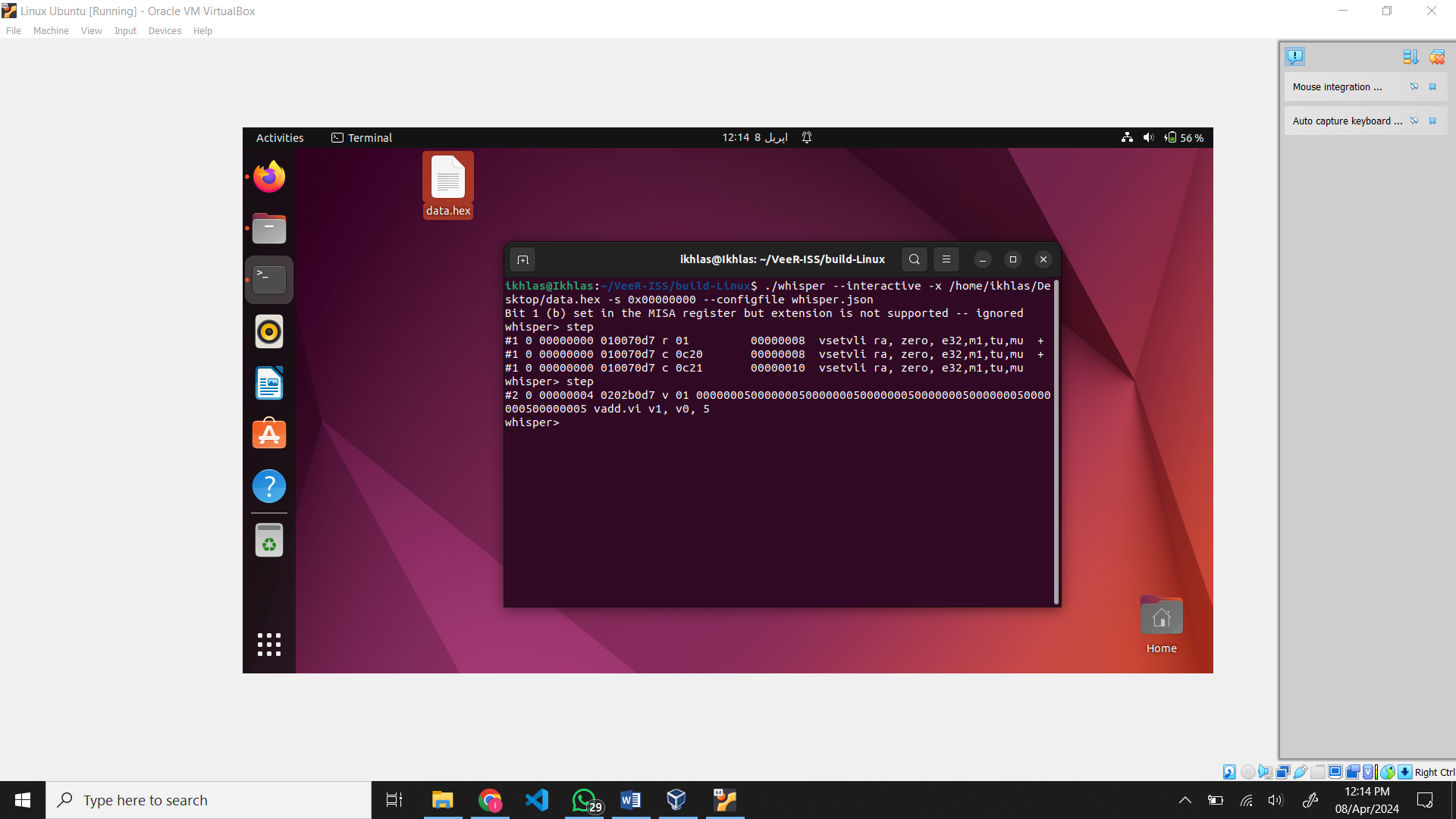
**Code to add 2 vectors v1 and v2 and store in vi.**

**OUTPUTS;**

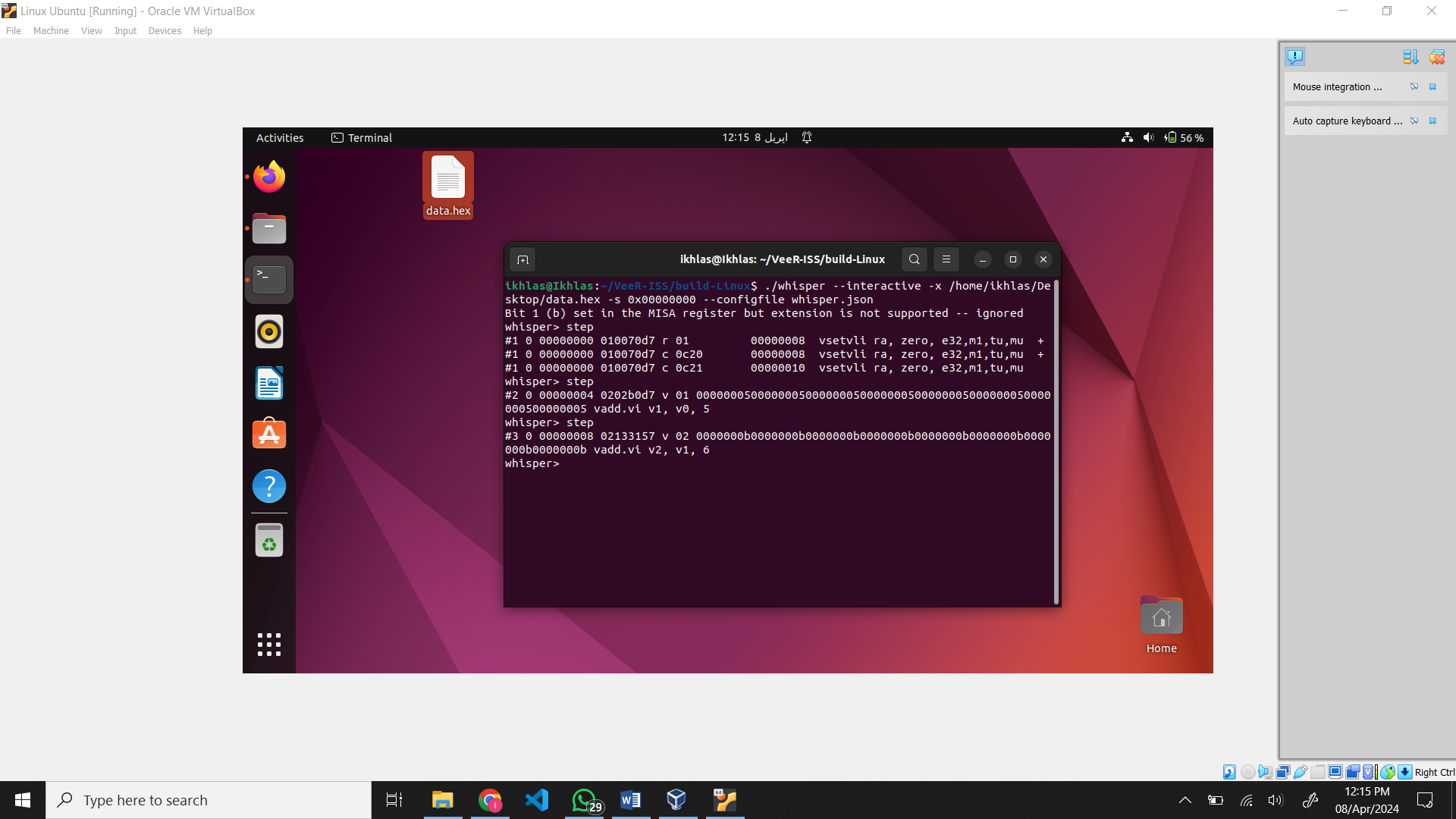
**FIRST INSTRUCTION EXECUTED**



SECOND INSTRUCTION EXECUTED



THIRD INSTRUCTION EXECUTION:



**SHORT NOTE:**

The key distinction between regular RISC-V instructions (like addi t1, t1, 5) and vector instructions (such as vadd.vi v1, v0, 0x5) lies in their handling of data. Regular RISC-V instructions are scalar, meaning they operate on one data element at a time. In contrast, vector instructions have the ability to process multiple data elements simultaneously. This capability makes vector instructions valuable for handling parallel data, often organized in vectors, arrays, or arrays of arrays resulting in faster computations.