A Simple VM for MIPS

Bhavye Jain

bhavyejain@ucsb.edu

MS 1st Year

Taanya Nithya Anand

taanya@ucsb.edu

MS 1st Year

Harshavardhan Miryala

harshavardhan@ucsb.edu

MS 1st Year

Overview

This project aims to build a virtual machine for the MIPS architecture. We will attempt to create a VM for a subset of the MIPS instruction set which allows us to run some interesting programs. Our pessimistic goals ensure we have a working VM that can execute some non-trivial programs and we hope to achieve our optimistic goal which will have the VM execute some useful programs. We discuss the programs that are expected to work on our VM which function as good test cases as well.

System Constraint Definition

Machine to Virtualize: MIPS64 architecture with the little-endian byte order

MIPS Subset:
addiu , sw, move, li, nop, addu, lw, lui, jal, jr, subu, beq, bne, slt, b

lwl, lwr,lbu, sb, lhu, sh (optimistic)

Directives: .ascii, .word

Restricted C Language:

Data types: int, float, char array (strings)

Branching: if-else with relational operators ==, !=, <, <=, >, >=

Loops: for, while

Libraries: stdio.h with only printf.

Deliverables

- **Pessimistic (basic functionality)**:
    - Implement VM for the MIPS subset
    - Support printf (and the stdio.h library)
    - Support non-array numeric operations
- **Optimistic (we aim to achieve this)**:
    - Support array operations (indirection)
    - Support user interaction through scanf

Success Criteria: The VM supports non-trivial program execution using assembly files created from a GCC cross-compiler for MIPS. I/O library calls should be supported by the VM.

Evaluation

C programs for testing:

1. Addition and subtraction of integers
2. Addition and subtraction of floating point numbers.
3. Swap 2 integers.
4. Swap 2 floats.
5. Find nth fibonacci number
6. Find min element in an array of integers.

Optimistic tests

7. Concatenate 2 strings
8. Find frequency of a char in a string
9. Program to find whether substring is present in the string
10. BubbleSort to sort an array of unsigned integers.

The input values are hardcoded. As part of optimistic goals, we will be taking input from the user using scanf. The outputs are printed to the terminal with printf.