

VM Get Started

[Will not count towards your grade]

Goal:

This milestone is simply to give you a chance to play with the target virtual machine for the CSC 453 Tau compiler that you are writing.

Specifications

The VM is distributed in the `tau` repository in the `vm` directory. The files are:

- `vm.py`: This file runs VM code.
- `vm_insns.py`: This file contains the VM instructions. **Use this file to find documentation on the VM instructions.**
- `vm_parser.py`: This file contains a parser for the VM assembly language textual representation. You should not need to examine this file.
- `vm_scanner.py`: This file contains a scanner for the VM assembly language textual representation. You should not need to examine this file.
- `vmcmd.py`: This file contains a command-line interface for the VM. You should not need to examine this file.

Command-line interface

To execute a VM program, `sample.vm`, you would type:

```
python3 -m tau.vm.vmcmd.py --file sample.vm    arg1 arg2 arg3 ...
```

The arguments are not needed for this milestone, but you may find them useful for future milestones because it will create a VM stack frame with the arguments, to be passed to your main routine.

If you invoke the command above with the `--verbose` option, you will see the VM instructions being executed.

Testing

I suggest you try to write the following VM programs:

- A program that adds/subtracts/multiplies/divides two numbers and prints the result.
- A program that uses the `Call` operation and successfully returns.
- A program that prints the numbers 0 through 9 in a loop.

Sample

```
loadi r1, 7
print r1
halt "end of program"
```

Difficulty

This milestone does not require a lot of code beyond the tree walker, which is provided. That said, it can be a little tricky to check and infer types for some nodes.

Start early and ask questions.

Turning in the program

There is nothing to turn in.