

# The Virtual Machine: Part 2

Prof. Naga Kandasamy  
ECE Department, Drexel University

Extend the VM translator developed in the previous assignment to add support for program-flow commands and subroutines. This problem is due December 4, 2023, by 11:59 pm. Please submit original work.

Extend your current VM translator implementation to add support for *if-goto* and *goto* commands for conditional and unconditional jumps as well as the following three function-related commands:

- function  $f\ k$ : This command starts the code of a function named  $f$  which has  $k$  local variables.
- call  $f\ n$ : This command calls function  $f$ , stating that  $n$  arguments have already been pushed onto the stack by the caller.
- return: This command returns to the calling function.

You may use the code provided in *vm\_translator\_v2.py* as a starting point for your solution. Edit the file as desired. The program takes as input two command-line arguments: *some-file.asm*, name of the Hack assembly file to be generated by the translator; and *path-name*, name of directory containing the *.vm* files. The generated *some-file.asm* file is placed in the same directory containing the *.vm* files.

```
> Python vm_translator_v2.py some-file.asm path-name
```

```
some-file.asm: assembly file generated by the translator  
path-name: directory containing the .vm source files
```

```
Example: Python vm_translator_v2.py mult-final.asm ./mult
```

## Testing Your VM Translator

Test the correctness of the assembly code generated by your translator using the following test programs. In each case, the expected output is specified within the comments in the *sys.vm* file. Use the Assembler tool to convert the generated *.asm* file into a *.hack* file containing the corresponding machine code. Then simulate the Hack machine code using the CPUEmulator tool and verify that the correct result is generated by the processor.

1. **(10 points)** The *add* directory contains functions that implement addition. Generate the assembly file as follows:

```
Python vm_translator_v2.py add-final.asm ./add
```

2. **(10 points)** The *arithmetic* directory contains functions that implement a sequence of addition and subtraction operations. Generate the assembly file as follows:

```
Python vm_translator_v2.py arithmetic-final.asm ./arithmetic
```

3. **(10 points)** The *mult* directory contains VM code that implement multiplication. Generate the assembly file as follows:

```
Python vm_translator_v2.py mult-final.asm ./mult
```

4. **(10 points)** The *factorial* directory contains VM functions that calculate the factorial of a given number. Generate the assembly file as follows:

```
Python vm_translator_v2.py factorial-final.asm ./factorial
```

5. **(10 points)** The *fibonacci* directory contains VM code to calculate the  $n^{\text{th}}$  element of the Fibonacci series in recursive fashion. Generate the assembly file as follows:

```
Python vm_translator_v2.py fibonacci-final.asm ./fibonacci
```

### Submission Instructions

Submit via BBLearn, the source code for the VM translator along with a README file that details how to build and execute it.